

Article

Phymastichus–Hypothenemus Algorithm for Minimizing and Determining the Number of Pinned Nodes in Pinning Control of Complex Networks

Jorge A. Lizarraga ¹, Alberto J. Pita ¹, Javier Ruiz-Leon ², Alma Y. Alanis ³, Luis F. Luque-Vega ^{4,*}, Rocío Carrasco-Navarro ⁵, Carlos Lara-Álvarez ⁶, Yehoshua Aguilar-Molina ⁷ and Héctor A. Guerrero-Osuna ⁸

- ¹ Departamento de Investigación, Centro de Enseñanza Técnica Industrial, Guadalajara 44638, Jalisco, Mexico; jlizarraga@ceti.mx (J.A.L.); ajaimes@ceti.mx (A.J.P.)
- ² CINVESTAV, Unidad Guadalajara, Guadalajara 45017, Jalisco, Mexico; javier.ruiz@cinvestav.mx
- ³ CUCEI, Universidad de Guadalajara, Guadalajara 44430, Jalisco, Mexico; alma.alanis@academicos.udg.mx
- ⁴ Department of Technological and Industrial Processes, ITESO, Tlaquepaque 45604, Jalisco, Mexico
- ⁵ Department of Mathematics and Physics, ITESO, Tlaquepaque 45604, Jalisco, Mexico; rociocarrasco@iteso.mx
- ⁶ Centro de Investigación en Matemáticas, Unidad Zacatecas, Zacatecas 98160, Zacatecas, Mexico; carlos.lara@cimat.mx
- ⁷ Departamento de Ciencias Computacionales e Ingenierías, Centro Universitario de los Valles de la Universidad de Guadalajara, Ameca 46600, Jalisco, Mexico; yehoshua.aguilar@academicos.udg.mx
- ⁸ Posgrado en Ingeniería y Tecnología Aplicada, Unidad Académica de Ingeniería Eléctrica, Universidad Autónoma de Zacatecas, Zacatecas 98000, Zacatecas, Mexico; hectorguerrero@uaz.edu.mx
- * Correspondence: luisluque@iteso.mx

Abstract

Pinning control is a key strategy for stabilizing complex networks through a limited set of nodes. However, determining the optimal number and location of pinned nodes under dynamic and structural constraints remains a computational challenge. This work proposes an improved version of the Phymastichus–Hypothenemus Algorithm—Minimized and Determinated (PHA-MD) to solve multi-constraint, hybrid optimization problems in pinning control without requiring a predefined number of control nodes. Inspired by the parasitic behavior of *Phymastichus coffea* on *Hypothenemus hampei*, the algorithm models each agent as a parasitoid capable of propagating influence across a network, inheriting node importance and dynamically expanding search dimensions through its “offspring.” Unlike its original formulation, PHA-MD integrates variable-length encoding and V-stability assessment to autonomously identify a minimal yet effective pinning set. The method was evaluated on benchmark network topologies and compared against state-of-the-art heuristic algorithms. The results show that PHA-MD consistently achieves asymptotic stability using fewer pinned nodes while maintaining energy efficiency and convergence robustness. These findings highlight the potential of biologically inspired, dimension-adaptive algorithms in solving high-dimensional, combinatorial control problems in complex dynamical systems.

Keywords: biological basis; complex network; energy-efficient control; heuristic optimization; pinning control; nonlinear optimization



Academic Editors: Sándor Szénási and Gábor Kertész

Received: 7 September 2025

Revised: 28 September 2025

Accepted: 30 September 2025

Published: 9 October 2025

Citation: Lizarraga, J.A.; Pita, A.J.; Ruiz-Leon, J.; Alanis, A.Y.; Luque-Vega, L.F.; Carrasco-Navarro, R.; Lara-Álvarez, C.; Aguilar-Molina, Y.; Guerrero-Osuna, H.A. Phymastichus–Hypothenemus Algorithm for Minimizing and Determining the Number of Pinned Nodes in Pinning Control of Complex Networks. *Algorithms* **2025**, *18*, 637. <https://doi.org/10.3390/a18100637>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, research on the analysis and control of complex networks has advanced rapidly [1]. The presence of complex networks in nature and society has sparked significant interest in this field [2], as they are integral to various systems in our daily lives [3]. For instance, the organization and function of a cell result from complex interactions

between genes, proteins, and other molecules [4]; the brain comprises a vast network of interconnected neurons [5]; social systems can be represented by graphs illustrating interactions among individuals [6]; ecosystems consist of species whose interdependencies can be mapped into food webs [7]; and large networked infrastructures, such as power grids and transportation networks, can be represented by graphs describing interactions among components [8,9].

In complex networks, a key challenge is ensuring control and stability while minimizing resource consumption. This requires selecting a subset of nodes, known as pinning nodes, to control the entire network. Pinning control achieves global stabilization by intervening in these selected nodes. The optimal selection of these nodes is crucial, with various centrality metrics, like Degree Centrality [10], Betweenness Centrality [11], and Closeness Centrality [12], proposed in the literature. These metrics offer different advantages depending on the network's structure and dynamics [3,13,14].

Recent advances in complex network control have led to more sophisticated algorithms designed to handle the growing complexity of modern networks. Techniques such as adaptive control [15], robust control [16], and sliding mode control [17] have been developed to improve stability and robustness under various uncertainties. However, these methods often struggle with the optimal selection of pinning nodes, especially in large networks with vast combinations of nodes.

Existing methods are limited by their reliance on fixed-dimension approaches and their difficulty adapting to dynamic network environments. These algorithms often require prior knowledge of the network's structure, which may not always be available, and can become computationally expensive as the network grows, making them impractical for real-time use.

The PHA-MD algorithm proposed in this work addresses these limitations by offering a novel approach to pinning control. Inspired by the parasitic relationship between the *Phymastichus coffea* wasp and the *Hypothenemus hampei* coffee borer, PHA-MD prioritizes the network's asymptotic stability while optimizing pinning node selection. Unlike its predecessor [18,19], PHA-MD autonomously determines the number of pinning nodes, avoiding the issues associated with fixed-dimension algorithms. This innovation allows PHA-MD to dynamically adapt to network changes, ensuring stability even in complex scenarios.

PHA-MD also leverages the V-stability tool [20] to ensure stability during optimization. Comparative simulations demonstrate PHA-MD's superior performance compared to several heuristic optimization algorithms, particularly in achieving network stability with fewer pinned nodes and efficient energy use.

The performance of PHA-MD is compared with other heuristic optimization algorithms, such as Ant Lion Optimizer (ALO) [21], Teaching–Learning-Based Optimization (TLBO) [22], Grey Wolf Optimizer (GWO) [23], Animal Migration Optimization (AMO) [24], Particle Swarm Optimization (PSO) [25], Artificial Bee Colony (ABC) [26], Gaining–Sharing Knowledge Based Algorithm (GSK) [27], Biogeography-Based Optimization (BBO) [28], Whale Optimization Algorithm (WOA) [29], Ant Colony Optimization (ACO) [30], Osprey Optimization Algorithm (OOA) [31], Mayfly Algorithm (MA) [32], Archimedes Optimization Algorithm (AOA) [33], Coronavirus Herd Immunity Optimizer (CHIO) [34], and Driving Training-Based Optimization (DTBO) [35].

These algorithms were selected for several reasons. They represent a wide range of optimization techniques, from bio-inspired methods like ALO [21] and PSO [25] to recent innovations such as CHIO [34] and OOA [31]. This diversity allows for a comprehensive comparison across different strategies. Many of these algorithms have shown high performance in solving complex optimization problems, making them robust benchmarks

for evaluating PHA-MD. Their popularity in the research community ensures that the comparison results are relevant and easily interpretable.

The selected algorithms also cover different optimization mechanisms, including swarm intelligence (PSO, WOA [29]), natural evolution (Genetic Algorithm (GA) [36]), intensive local search (GWO [23]), and reinforcement learning (Q-Learning based algorithms [37]). Comparing PHA-MD with these approaches helps its competitiveness with the latest developments in heuristic optimization.

In summary, the selection of these algorithms provides a rigorous benchmarking framework, allowing for a thorough assessment of PHA-MD's capabilities, particularly in solving the permutation problem in node selection and ensuring the network's asymptotic stability. Additionally, PHA-MD's ability to handle multi-constraint optimization problems makes it versatile and effective in various applications.

The rest of this paper is organized as follows. Section 2 provides an overview of complex network pinning control, detailing the V-stability tool and other essential mathematical preliminaries. This section also defines the optimization problem, focusing on minimizing the energy consumed by control actions at the pinning nodes and the number of pinning nodes required for network stability. In Section 3, the biological foundation of the PHA-MD algorithm is described, highlighting its unique characteristics and mechanisms. Section 4 presents a comprehensive simulation study of the proposed PHA-MD algorithm across various complex network topologies, including a detailed comparison with other heuristic optimization algorithms. Finally, Section 5 discusses the conclusions and potential future directions of the research.

2. Problem Statement

The optimization problem in complex networks is formulated to achieve two primary objectives:

- (i) *Minimizing the energy consumed by the control actions at the pinning nodes:* this objective focuses on reducing the overall energy required to stabilize the network through targeted interventions at selected nodes.
- (ii) *Minimizing the number of pinning nodes required to achieve network stability:* this objective aims to reduce the number of nodes that need to be pinned to maintain the stability of the entire network, thereby optimizing resource usage.

These objectives define the optimization problem as follows. Let $M \equiv Q(\Theta, G, K)$ denote the closed-loop matrix built from the node passivity degrees Θ , the coupling matrix G , and the diagonal gain matrix K ; let $\lambda\{\cdot\}$ denote the spectrum (set of eigenvalues); and let $R_e\{\cdot\}$ denote the real-part operator applied to those eigenvalues. Then

$$\begin{aligned} & \text{minimize} && \{ f(\mathcal{U}), \text{card}(\mathcal{U}) \} \\ & \text{subject to} && \mathcal{U} \subseteq \{1, \dots, N_n\}, \\ & && \max \{ R_e\{\lambda\{M\}\} \} < 0, \end{aligned} \quad (1)$$

where \mathcal{U} is the index set of pinning nodes and $\text{card}(\mathcal{U})$ is its cardinality. The control law for node n_i is $\mathbf{u}_i = -k_i \mathbf{x}_i$, with $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N_s})^T \in \mathbb{R}^{N_s}$ and $k_i \geq 0$ the i -th diagonal entry of K , defined as

$$K_{i,i} = \begin{cases} k_i, & i \in \mathcal{U}, \\ 0, & i \notin \mathcal{U}. \end{cases}$$

The inequality $\max R_e\{\lambda\{M\}\} < 0$ expresses exponential stability.

Specifically, the optimization problem involves the construction of the control gain matrix K . The goal is to find the optimal set of pinning nodes that achieves efficient network

control. In this context, the construction of the control gain matrix K is crucial. The matrix K determines the effectiveness of the pinning control strategy, directly impacting the stability and efficiency of the network. Ensuring an optimal configuration of K is essential for achieving network stability with minimal resource consumption. Thus, the following describes the relevance of the matrix K in the pinning control of complex networks, and its impact on the stability of the network.

Consider the following Network (2) of N_n nodes with linear diffusive couplings and an N_s -dimensional dynamical system [38]:

$$\dot{\mathbf{x}}_i = f_i(\mathbf{x}_i) + \sum_{j=1, j \neq i}^{N_n} c_{ij} a_{ij} \Gamma(\mathbf{x}_j - \mathbf{x}_i), \quad (2)$$

where $f_i : \mathbb{R}^{N_s} \rightarrow \mathbb{R}^{N_s}$ denotes the *self-dynamics* of node n_i , i.e., the intrinsic vector field that governs its evolution in isolation (all couplings and controls set to zero). The constants c_{ij} are the coupling strengths between n_i and n_j , $\Gamma \in \mathbb{R}^{N_s \times N_s}$ specifies how state components are coupled for each connected pair $(\mathbf{x}_j, \mathbf{x}_i)$, and the connection matrix $A = [a_{ij}] \in \mathbb{R}^{N_n \times N_n}$ encodes the network topology: if there is a connection between n_i and n_j for $i \neq j$, then $a_{ij} = a_{ji} = 1$; otherwise $a_{ij} = a_{ji} = 0$ for $i \neq j$. The diagonal elements are defined by $a_{ii} = -\sum_{j=1, j \neq i}^{N_n} a_{ij}$, where d_i is the degree of node n_i . A small subset of nodes in the network is subjected to local feedback as part of the pinning control, and these are called *pinning nodes* [39]. Assuming the diffusive condition

$$c_{ii} a_{ii} + \sum_{j=1, j \neq i}^{N_n} c_{ij} a_{ij} = 0, \quad i = 1, 2, \dots, N_n, \quad (3)$$

Network (2) can be rewritten in a compact controlled form as

$$\dot{\mathbf{x}}_i = f_i(\mathbf{x}_i) + \sum_{j=1}^{N_n} c_{ij} a_{ij} \Gamma \mathbf{x}_j + B_i \mathbf{u}_i, \quad (4)$$

where $B_i \in \mathbb{R}^{N_s \times N_s}$ is the *input (actuation) matrix* that maps the control vector $\mathbf{u}_i \in \mathbb{R}^{N_s}$ into the state derivatives of node n_i . In the context of pinning control, B_i specifies which state components are directly actuated (full or partial channels). Typical choices are $B_i = I_{N_s}$ (full-state actuation) or $B_i = \text{diag}(b_i)$ with $b_i \in \{0, 1\}^{N_s}$ (selected coordinates), where I_{N_s} denotes the $N_s \times N_s$ identity matrix (ones on the diagonal and zeros elsewhere). Unless otherwise stated, we adopt $B_i = I_{N_s}$ in the simulations.

The control input is defined nodewise as

$$\mathbf{u}_i(\mathbf{x}_i) = \begin{cases} -k_i \mathbf{x}_i, & i \in \mathcal{U}, \\ 0, & i \notin \mathcal{U}, \end{cases} \quad (5)$$

where \mathcal{U} is the set of pinning nodes with $1 \leq \text{card}(\mathcal{U}) \leq N_n$ [40]. Hence, the self-dynamics of the controlled nodes becomes

$$\dot{\mathbf{x}}_i = f_i(\mathbf{x}_i) - k_i B_i \mathbf{x}_i, \quad i \in \mathcal{U}. \quad (6)$$

To demonstrate network stability and to calculate the lower bound $N_{\lambda+}$ for the number of pinned nodes, consider a continuously differentiable Lyapunov function $V(\mathbf{x}_i) : D \subseteq$

$\mathbb{R}^{N_s} \mapsto \mathbb{R}_+$, satisfying $V(\bar{\mathbf{x}}) = 0$ with $\bar{\mathbf{x}} \in D$, such that for the self-dynamics (6), there is a scalar θ_i and k_i guaranteeing

$$\frac{\partial V(\mathbf{x}_i)}{\partial \mathbf{x}_i} (f_i(\mathbf{x}_i) - k_i B \mathbf{x}_i + \theta_i \Gamma \mathbf{x}_i + k_i \Gamma \mathbf{x}_i) < 0 \quad \forall \mathbf{x}_i \in D_i \subseteq D, \mathbf{x}_i \neq 0, \quad (7)$$

where

$$D_i = \{\mathbf{x}_i : \|\mathbf{x}_i - \bar{\mathbf{x}}_i\| < \alpha\}, \alpha > 0, D = \bigcup_{i=1}^{N_n} D_i,$$

and θ_i is the passivity degree of $f(\mathbf{x}_i)$. Then, to determine whether Network (4) is synchronized at the equilibrium point $\bar{\mathbf{X}} = (\bar{\mathbf{x}}_1^T, \dots, \bar{\mathbf{x}}_{N_n}^T)^T$, such that

$$\mathbf{x}_1(t) = \mathbf{x}_2(t) = \dots = \mathbf{x}_{N_n}(t) \rightarrow \bar{\mathbf{x}} \text{ as } t \rightarrow \infty,$$

let us consider

$$V_N(\mathbf{X}) = \sum_{i=1}^{N_n} \frac{1}{2} \mathbf{x}_i^T \mathbf{P} \mathbf{x}_i, \quad \mathbf{P} = \mathbf{P}^T > 0, \quad (8)$$

for the controlled Network (4); then

$$\begin{aligned} \dot{V}_N(\mathbf{X}) &= \sum_{i=1}^{N_n} \mathbf{x}_i^T \mathbf{P} \left(f_i(\mathbf{x}_i) + \sum_{j=1}^{N_n} c_{ij} a_{ij} \Gamma \mathbf{x}_j - k_i B \mathbf{x}_i \right) \\ \dot{V}_N(\mathbf{X}) &< \sum_{i=1}^{N_n} \mathbf{x}_i^T \mathbf{P} \left(\theta_i \Gamma \mathbf{x}_i + \sum_{j=1}^{N_n} c_{ij} a_{ij} \Gamma \mathbf{x}_j + k_i \Gamma \mathbf{x}_i \right) \end{aligned} \quad (9)$$

or, using the Kronecker product,

$$\dot{V}_N(\mathbf{X}) < \mathbf{X}^T (-\Theta + G - K) \otimes \mathbf{P} \Gamma \mathbf{X}, \quad (10)$$

where $c_{ij} a_{ij}$ are the entries of $G \in \mathbb{R}^{N_n \times N_n}$, $\Theta = \text{diag}(\theta_1, \theta_2, \dots, \theta_{N_n})$, $K = \text{diag}(k_1, k_2, \dots, k_{N_n})$, and $\mathbf{P} \Gamma \geq 0$. Then, according to [20], Network (2) is locally asymptotically stable around its equilibrium point if the closed-loop characteristic matrix

$$Q = -\Theta + G - K \quad (11)$$

is negative definite; then, the number of pinned nodes cannot be less than the number of positive eigenvalues N_{λ^+} with $K = [0]^{N_n \times N_n}$, such that, $\text{card}(\mathcal{U}) > N_{\lambda^+}$. Thus, satisfying the above conditions, Network (2) is V-stable [20], demonstrating the importance of the K matrix to ensure stability in fixation control.

In (11), the importance of the K matrix for ensuring stability in the pinning control is evident. However, the energy consumed by the pinning nodes during the control process is a significant factor that impacts the overall efficiency of the network. Therefore, optimizing the energy consumption is essential to ensure that the network operates effectively without unnecessary expenditure of resources.

The objective function $f(\mathcal{U})$ in (1) focuses on minimizing the energy consumed by the pinning nodes at a specific time t_0 . This approach not only helps in achieving the desired control with minimal energy, but also ensures that the network remains stable and efficient. By concentrating on energy optimization, the control strategy can be made more sustainable and cost-effective, addressing both performance and resource utilization concerns in complex network systems.

The selected objective function for this problem is the energy consumed by the pinning nodes at time t_0 , $E_{t_0}(\mathcal{U})$, defined as

$$E_{t_0}(\mathcal{U}) = \frac{1}{2} \sum_{i \in \mathcal{U}} \|k_i \mathbf{x}_i(t_0)\|^2, \quad (12)$$

where \mathcal{U} is the set of pinned nodes (only those that contribute to the sum; equivalently, summing over all i with $K_{ii} = 0$ for $i \notin \mathcal{U}$ yields the same value).

Remark 1. Considering V -stability and that energy is consumed only by the pinning nodes, the maximum energy occurs at time t_0 [20]. This allows the optimization to jointly minimize $E_{t_0}(\mathcal{U})$ and $\text{card}(\mathcal{U})$, ensuring efficient and stable control of the complex network.

3. PHA-MD Algorithm

The PHA-MD algorithm iteratively constructs the control gain matrix K , leveraging the adaptive nature of its agents. Each agent presents a unique and variable candidate solution in each iteration, inheriting information and discarding non-optimal nodes to improve the solution in successive generations. This approach enables the optimization of the pinning node set configuration \mathcal{U} , achieving a network controlled efficiently with the minimum number of pinning nodes and the lowest possible energy consumption. In summary, the PHA-MD algorithm excels in simultaneously optimizing the energy consumed and the cardinality of the set of pinning nodes, optimizing the control gain matrix K for complex networks.

3.1. Biological Basis

The proposed PHA-MD algorithm is inspired by the parasitic symbiotic relationship between the parasitoid wasp (*Phymastichus coffea*, LaSalle, Hymenoptera: Eulophidae, Figure 1a) and the coffee borer (*Hypothenemus hampei*, Ferrari, Coleoptera: Curculionidae: Scolytinae, Figure 1b). This symbiotic relationship is used as a biological control mechanism because the borer is considered the most harmful pest affecting the coffee crop due to its attack on the berry, producing weight loss, depreciation of the grain, and loss of quality due to the presence of impurities in the infected beans [41].

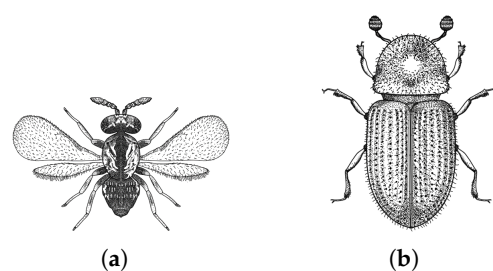


Figure 1. (a) *Phymastichus coffea* Lasalle (*P. coffea*). (b) *Hypothenemus hampei* Ferrari (*H. hampei*).

The life cycle of the wasp begins when *P. coffea* parasitizes *H. hampei*, depositing up to two eggs per host, typically one male and one female. From this point, the incubation process begins. Upon hatching, the larvae feed on the abdominal tissues of the host, the coffee borer, until they complete their metamorphosis. Once this stage is finished, the larvae emerge from the host as adult wasps [42].

The i -th symbiotic relationship $\phi_i(t)$ between the i -th agents *P. coffea* and *H. hampei* at iteration t is represented as

$$\phi_i(t) = \{F_i(t), h_i(t), \mathbf{p}_i(t), \mathbf{h}_i(t)\}, \quad (13)$$

where $\mathbf{p}_i \in \mathbb{N}^{j \in [1, N_n]}$ and $\mathbf{h}_i \in \mathbb{R}^{N_n}$ are the i -th agents *P. coffea* and *H. hampei* respectively, whose populations are defined as

$$P(t) = \begin{Bmatrix} \mathbf{p}_1(t) \\ \vdots \\ \mathbf{p}_{N_a}(t) \end{Bmatrix}, \quad H(t) = \begin{bmatrix} \mathbf{h}_1(t) \\ \vdots \\ \mathbf{h}_{N_a}(t) \end{bmatrix}, \quad (14)$$

where N_a is the number of agents for both populations. $F_i(t)$ is a penalty function, defined as

$$F_i(t) = E_{t_0}(\mathcal{U}) + \alpha_1 c_i^{(1)}(t) + \alpha_2 \Lambda(\mathbf{h}_i(t)), \quad (15)$$

where α_1 and α_2 are penalty parameters, with the difference between them defining the importance between the second and third terms in (15); $c_i^{(1)} = \text{card}(\mathbf{p}_i)$ is the cardinality of the agent $\mathbf{p}_i(t)$; and $\Lambda(\mathbf{h}_i(t))$ is the maximum value between zero and the maximum real eigenvalue of matrix (11), defined as follows:

$$\Lambda(\mathbf{h}_i(t)) = [\max\{R_e\{\lambda\{Q(\Theta, G, K_i)\}\}\}]^+, \quad (16)$$

where the elements $k_{i,j}$ of the main diagonal of the i -th control gains matrix K_i are determined by the relationship of the agents \mathbf{p}_i and \mathbf{h}_i , such that

$$k_{i,j} = \begin{cases} h_{i,j} & j \in \mathbf{p}_i \\ 0 & j \notin \mathbf{p}_i. \end{cases} \quad (17)$$

The symbiotic relationship ϕ is classified in three possible favorable events for the *P. coffea* species (see Figure 2), which are as follows:

$$\phi_i^{(1)}(t) = \{F_i^{(1)}(t), \Lambda_i^{(1)}(t), \mathbf{p}_i^{(1)}(t), \mathbf{h}_i^{(1)}(t)\}, \quad (18)$$

$$\phi^{(2)}(t) = \{F^{(2)}(t), \Lambda^{(2)}(t), \mathbf{p}^{(2)}(t), \mathbf{h}^{(2)}(t)\}, \quad (19)$$

$$\phi^{(3)}(t) = \{F^{(3)}(t), \Lambda^{(3)}(t), \mathbf{p}^{(3)}(t), \mathbf{h}^{(3)}(t)\}, \quad (20)$$

where $\phi_i^{(1)}(t)$ is the favorable parasitization event, when the wasp parasites the beetle; $\phi^{(2)}(t)$ implies the birth of only a female; and $\phi^{(3)}(t)$ corresponds to the birth of a female and a male. Note that types 2 and 3 affect the entire population; however, type 1 is particular for each agent, and this is reflected in the subscript i for this type of symbiotic relationship. A single male fertilizes numerous females; however, it requires a female to bore the host abdomen in order to emerge, and therefore, the birth of a female and a male is the ideal scenario for the *P. coffea* species [42].

The concept of *Memory Inheritance* justifies the relevance of these events. This attribute implies that each i -th wasp can pass down two types of memories, $\mathbf{m}_i \in \mathbb{N}^{N_n}$ and $\mathbf{w}_i \in \mathbb{R}^{N_n}$, to its future generations. Since the coffee tree can be modeled as a graph, whose berries are equivalent to nodes, \mathbf{m} and \mathbf{w} are, respectively, “how many nodes to select” and “which nodes to select”.

The example in Figure 3 shows the decision of the i -th agent $\mathbf{p}_i(t)$ to visit four nodes, selecting n_{18} , n_{15} , n_{10} , and n_8 . The elements $m_{i,j}$ and $w_{i,j}$ of each memory are selected by the roulette method. Note that in case of $m_{i,j}(t) \neq j$ or $w_{i,j}(t) \neq i$, if the element is selected, it returns the index it represents, and not its occurrence value. In the following sections, the process of modifying the frequency of occurrence is explained.

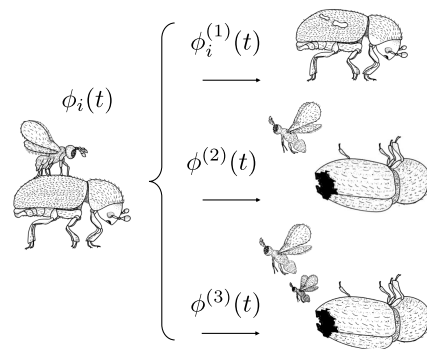


Figure 2. Parasitic symbiotic relationship between agents, favorable events. (1) Parasitization, (2) birth of a female wasp, (3) birth of a female and a male wasp.

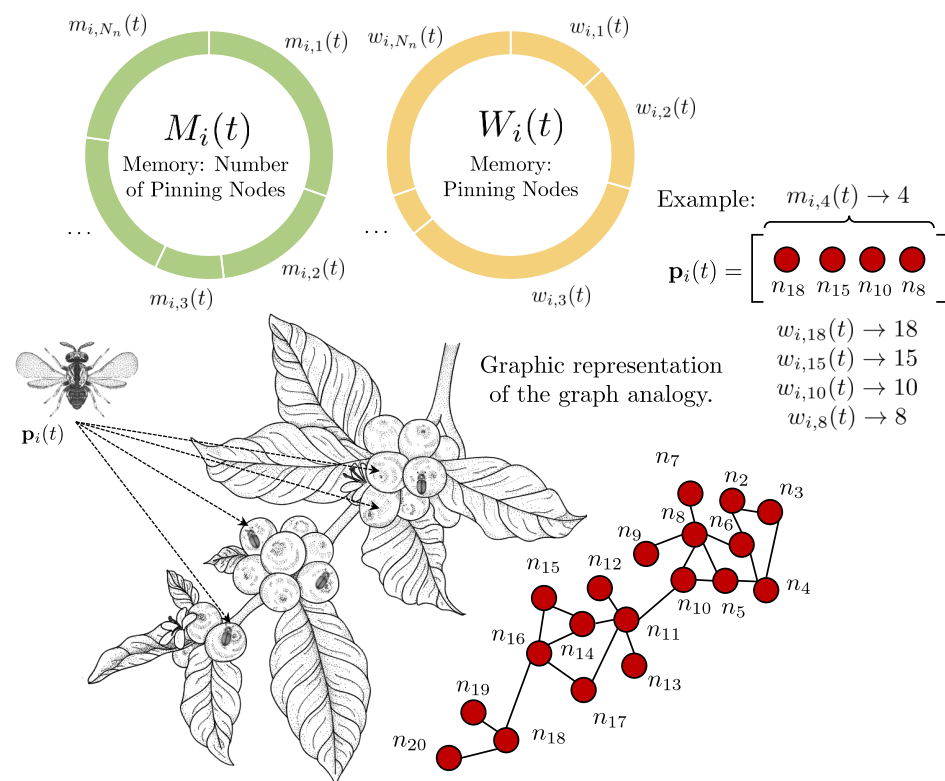


Figure 3. Behavior and modeling of *P. coffea* wasp: visits of *P. coffea* wasp, coffee tree graph.

Another attribute of biological inspiration is the behavior of *H. hampei*. In nature, the beetle drills into coffee berries at or near the apex, which is the softest area of the coffee bean [43] (Figure 4). Similar to how the gravity, shape, relief, or orientation of the berry influence its movement, the *borer functions* guide *H. hampei* towards the optimal regions identified by the group of beetles, without direct interference. These functions reduce the search space in favor of the symbiotic relationship $\phi^{(2)}$, because the beetles that manage to bore into the fruit will be able to feed, thereby increasing the probability that at least one *P. coffea* individual will be born. The *borer functions* are defined as

$$\beta^{(1)} = \mathbf{h}^{(2)} - \zeta|\mathbf{h}^{(2)}| + \left(\beta^{(1)}(0) - \mathbf{h}^{(2)} + \zeta|\mathbf{h}^{(2)}|\right)e^{\left(\frac{-\gamma\tau}{N_\tau}\right)}, \quad (21)$$

$$\beta^{(2)} = \mathbf{h}^{(2)} + \zeta|\mathbf{h}^{(2)}| - \left(\mathbf{h}^{(2)} - \beta^{(2)}(0) + \zeta|\mathbf{h}^{(2)}|\right)e^{\left(\frac{-\gamma\tau}{N_\tau}\right)}, \quad (22)$$

where $\beta^{(1)}$ and $\beta^{(2)}$ are the lower and upper borer functions, respectively; $\beta_j^{(1)} \leq h_{i,j} \leq \beta_j^{(2)} \forall j \in [1, N_n]$, $|\cdot|$ denotes the absolute value; ζ is the compression parameter, which prevents the limits $\beta^{(2)}$ and $\beta^{(1)}$ from converging to $\mathbf{h}^{(2)}$; N_τ is the maximum interaction number per epoch; γ is an iteration coefficient; and $\beta^{(1)}(0)$ and $\beta^{(2)}(0)$ are, respectively, the initial upper and lower bounds. It is important to mention that the *borer functions* limit the values of each dimension j ; however, they can be specific for each i -th agent *H. hampei*. It should be understood that with the notation of (21) and (22), the entire population H is delimited equally.

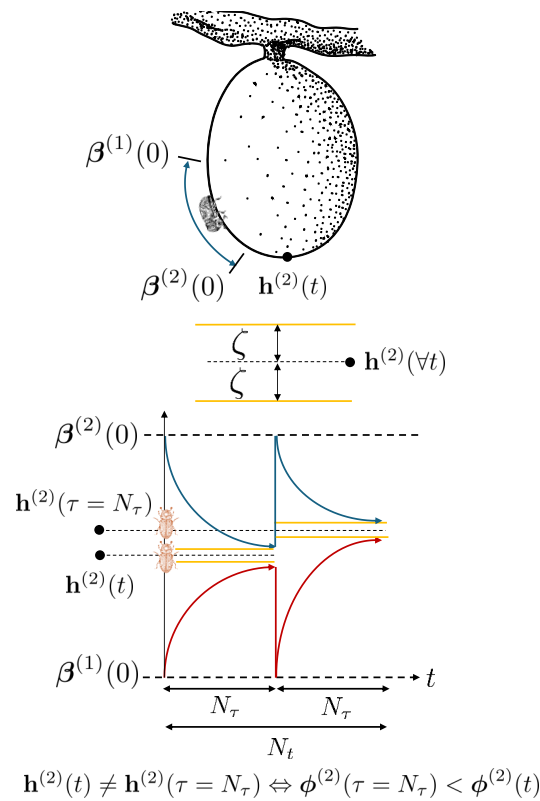


Figure 4. The biological inspiration for the borer functions: the path of *Hypothenemus hampei* towards the coffee berry apex.

3.2. Attribute Updating

This section describes the creation of new generations for each type of agent based on updating parameters such as symbiotic relationships, updating subtypes, and optimizing inheritance memories. The subtypes of symbiotic relationships are updated as follows:

$$\phi_i^{(1)}(t) = \begin{cases} \phi_i(t) & \text{if } t = 1 \vee \tau = 1 \\ \phi_i(t) & \text{if } t > 1 \wedge \tau > 1 \wedge F_i(t) < F_i^{(1)}(t-1) \\ \phi_i^{(1)}(t-1) & \text{else,} \end{cases} \quad (23)$$

$$\phi^{(2)}(t) = \begin{cases} \phi_{\min}^{(1)}(t) & \text{if } t = 1 \vee \tau = 1 \\ \phi_{\min}^{(1)}(t) & \text{if } t > 1 \wedge \tau > 1 \wedge F_{\min}^{(1)}(t) < F^{(2)}(t-1) \\ \phi^{(2)}(t-1) & \text{else,} \end{cases} \quad (24)$$

$$\phi^{(3)}(t) = \begin{cases} \phi^{(2)}(t) & \text{if } \tau = N_\tau \wedge \mathcal{B}(t) = \emptyset \wedge h^{(2)}(t) = 0 \\ \phi^{(2)}(t) & \text{if } \tau = N_\tau \wedge \mathcal{B}(t) \neq \emptyset \wedge h^{(2)}(t) = 0 \wedge c^{(2)}(t) < c^{(3)}(t-1) \\ \phi^{(2)}(t) & \text{if } \tau = N_\tau \wedge \mathcal{B}(t) \neq \emptyset \wedge h^{(2)}(t) = 0 \wedge c^{(2)}(t) = c^{(3)}(t-1) \\ & \wedge F^{(2)}(t) < F^{(3)}(t-1) \\ \phi^{(3)}(t-1) & \text{else,} \end{cases} \quad (25)$$

where τ is the epoch trigger, which increases by one with each iteration t . When $\tau = N_\tau$, a new epoch starts, whose first generation has its initial conditions reset, including the epoch trigger (Figure 4), $F_{\min}^{(1)}(t) = \min_{i \in [1, N_a]} \{F_i^{(1)}(t)\}$.

Figure 5 details how PHA-MD updates the three symbiotic subtypes at each iteration and at the end of every epoch. First, each agent evaluates its candidate and updates the per-agent best $\phi_i^{(1)}(t)$ whenever its current cost $F_i(t)$ improves (Equation (23)). Next, the algorithm promotes the population-wide best $\phi^{(2)}(t)$ using the minimum among $\{F_i^{(1)}(t)\}$ (Equation (24)). When the epoch boundary is reached ($\tau = N_\tau$), the blocking stage is invoked; based on stability ($h^{(2)} = 0$), cardinality, and cost, $\phi^{(2)}(t)$ may be adopted as the epoch-level best $\phi^{(3)}(t)$ (Equation (25)). If the stated conditions are not met, the corresponding records are preserved (“else” branches in Equations (23)–(25)). While the flow focuses on ϕ -updates, it is coordinated with the blocking-set dynamics $\mathcal{B}(t)$ (Equations (26) and (27)) that are executed at epoch end.

In the PHA-MD algorithm, a new feature has been introduced compared to its previous version, a node-blocking stage. This stage involves the identification and blocking of nodes, with the set of blocked nodes denoted as $\mathcal{B}(t)$. During each epoch, if and only if $h^{(1)}(t) = 0$, a node will be selected for blocking. This action increases the cardinality of the set $\mathcal{B}(t)$ by one. The set $\mathcal{B}(t)$ captures the nodes that are currently blocked at time t , reflecting the evolving state of the algorithm as it progresses through each epoch. The set of blocked nodes is defined as

$$\mathcal{B}(t) = \begin{cases} \emptyset & \text{if } t < N_\tau \\ b(t) & \text{if } \tau = N_\tau \wedge b(t) = \emptyset \wedge h^{(2)}(t) = 0 \\ \mathcal{B}(t-1) \cup b & \text{if } \tau = N_\tau \wedge \mathcal{B}(t) \neq \emptyset \wedge h^{(2)}(t) = 0 \wedge N_n - c^{(\mathcal{B})} > N_{\lambda+} \\ \mathcal{B}_{\max}(t) & \text{if } \tau = N_\tau \wedge \mathcal{B}(t) \neq \emptyset \wedge h^{(2)}(t) \neq 0 \\ \mathcal{B}(t-1) & \text{else,} \end{cases} \quad (26)$$

where $b(t)$ is a blocked node, and $\mathcal{B}_{\max}(t)$ is the best set of blocked nodes, such that

$$\mathcal{B}_{\max}(t) = \begin{cases} \emptyset & \text{if } t < N_\tau \\ \mathcal{B}(t-1) & \text{if } \tau = N_\tau \wedge \mathcal{B}(t) = \emptyset \wedge h^{(2)}(t) = 0 \\ \mathcal{B}(t-1) & \text{if } \tau = N_\tau \wedge \mathcal{B}(t) \neq \emptyset \wedge h^{(2)}(t) = 0 \wedge c^{(\mathcal{B})} \geq c^{(\mathcal{B}_{\max})} \\ \mathcal{B}_{\max}(t-1) & \text{else,} \end{cases} \quad (27)$$

with $c^{(\mathcal{B})} = \text{card}(\mathcal{B})$ and $c^{(\mathcal{B}_{\max})} = \text{card}(\mathcal{B}_{\max})$. The i -th node will be blocked when its number of visits implies the minimum number of successes achieved, thus

$$b(t) = \max_{i \in [1, N_n]} \left\{ \sum_{j=1}^{N_a} \frac{v_{i,j}(t)}{w_{i,j}(t)} \right\}, \quad (28)$$

where $v_{i,j}(t)$ represents the number of visits made by the i -th agent of *P. coffea* at the j -th node. A visit by this agent is considered successful if it leads to performance improve-

ment. The memory vector $\mathbf{w}_i(t) \in \mathbb{R}^{1 \times N_n}$ increases its j -th dimension by 1 to signify successful visits.

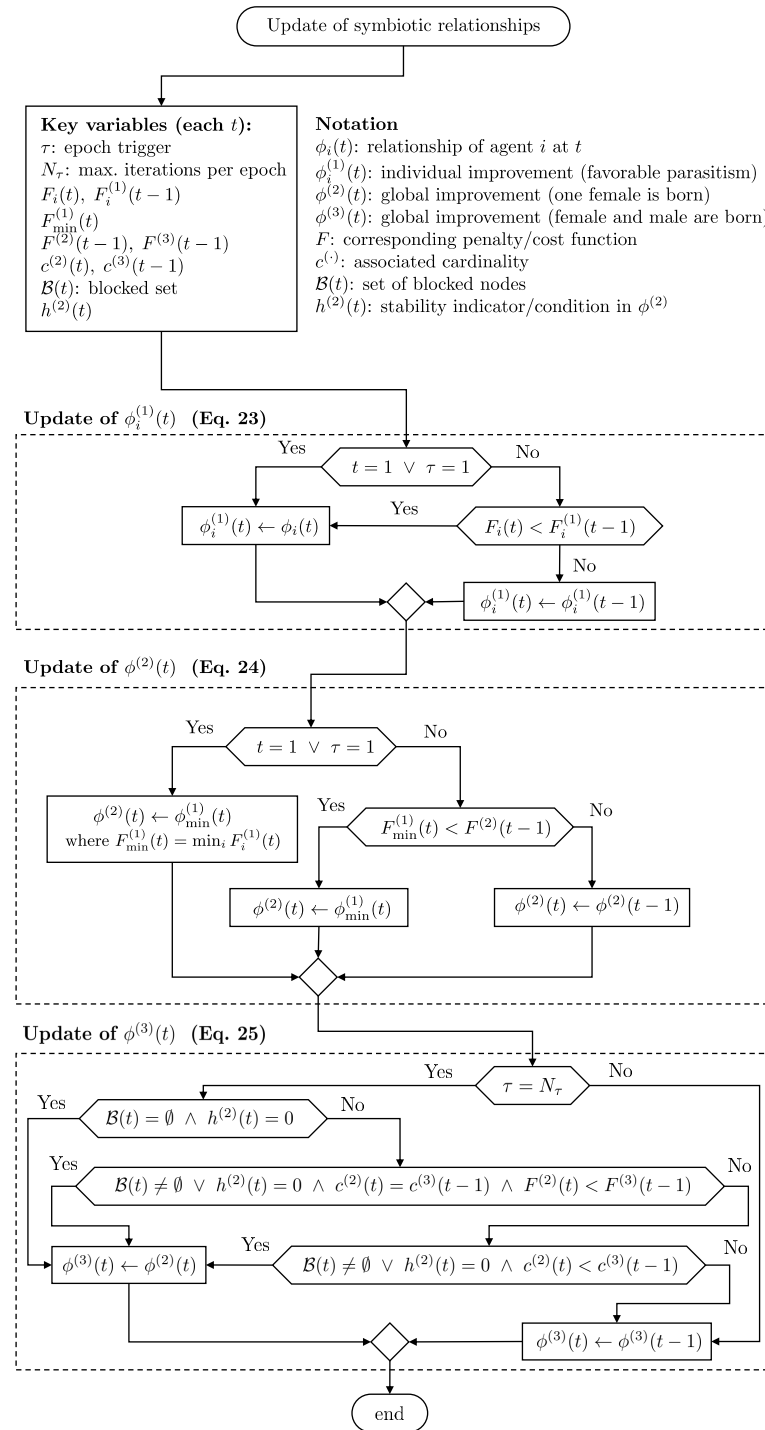


Figure 5. A flowchart for the updating of symbiotic relationships in PHA-MD. The diagram summarizes the decision logic for the three favorable subtypes: per-agent best $\phi_i^{(1)}(t)$ (Equation (23)), population best $\phi^{(2)}(t)$ (Equation (24)), and epoch-level best under blocking $\phi^{(3)}(t)$ (Equation (25)), with the epoch trigger τ and the blocking set $\mathcal{B}(t)$ governing transitions at $\tau = N_\tau$.

In this context, $v_{i,j}(t)$ accounts for both successful and unsuccessful visits, while $w_{i,j}(t)$ specifically tracks successful visits by incrementing its j -th dimension when the i -th agent enhances its performance after visiting the j -th node. As shown in Figure 3, the first type of memory that *P. coffea* agents inherit from their offspring is $M \in \mathbb{R}^{N_a \times N_n}$. This

memory represents “how many nodes to select”, that is, the cardinality of each agent, and its elements $m_{i,j}(t)$ are updated in the following way:

$$m_{i,j}(t+1) = \begin{cases} 1 \forall i & \text{if } t = 1 \wedge j > N_{\lambda+} \\ 0 \forall i & \text{if } t = 1 \wedge j \leq N_{\lambda+} \\ m_{i,j}(t) + 1 & \text{if } F_i^{(1)}(t) < F_i^{(1)}(t-1) \wedge j = c^{(1)}(t) \\ 0 \forall i & \text{if } h_i^{(1)}(t) = 0 \wedge j > c^{(1)}(t) \\ 1 \forall i & \text{if } \tau = N_{\tau} \wedge j \in [N_n - c^{(B)} - 1, N_n - c^{(B)}] \\ 0 \forall i & \text{if } \tau = N_{\tau} \wedge j \notin [N_n - c^{(B)} - 1, N_n - c^{(B)}] \\ m_{i,j}(t) & \text{else,} \end{cases} \quad (29)$$

where $c^{(1)}(t) = \text{card}(\mathbf{p}_i^{(1)}(t))$. The second type of memory that *P. coffea* agents inherit from their offspring is $W \in \mathbb{R}^{N_a \times N_n}$. This memory represents “which nodes to select”, and its elements $w_{i,j}(t)$ are updated in the following way:

$$w_{i,j}(t+1) = \begin{cases} 1 & \text{if } t = 1 \\ w_{i,j}(t) + 1 & \text{if } F_i^{(1)}(t) < F_i^{(1)}(t-1) \wedge j \in \mathbf{p}_i^{(1)}(t) \\ w_{i,j}(t) + 1 & \text{if } F_i^{(2)}(t) < F_i^{(2)}(t-1) \wedge i \in \mathcal{R}(t) \wedge j \in \mathbf{p}^{(2)}(t) \\ 0 \forall i & \text{if } \tau = N_{\tau} \wedge j \in \mathcal{B}(t) \\ 1 \forall i & \text{if } \tau = N_{\tau} \wedge j \notin \mathcal{B}(t) \\ w_{i,j}(t) & \text{else,} \end{cases} \quad (30)$$

where $\mathcal{R}(t)$ is a set of uniformly distributed random nodes (without repetition) with random cardinality.

3.3. Saturated Sigmoid Switch Functions

Saturated sigmoids are employed as compact binary comparators (switches) with explicit tie handling through a small $\epsilon > 0$. The parameter ϵ is taken as a fixed tolerance (machine epsilon, *eps*) and is used to disambiguate equalities [44]. Let $x = (a - b)$ denote the comparison residual. The ideal steep-slope limit is indicated by “ $\cdot\infty$ ” (in code, a large gain $g \gg 1$ is used).

$$S_1(x) = \frac{1}{1 + e^{(x+\epsilon)\cdot\infty}} = \begin{cases} 1, & x < 0, \\ 0, & x > 0, \\ 0, & x = 0, \end{cases} \quad \text{“<” switch.} \quad (31)$$

$$S_2(x) = 1 - \frac{1}{1 + e^{(x-\epsilon)\cdot\infty}} = \begin{cases} 1, & x > 0, \\ 0, & x < 0, \\ 0, & x = 0, \end{cases} \quad \text{“>” switch.} \quad (32)$$

$$S_3(x) = \frac{1}{1 + e^{(x-\epsilon)\cdot\infty}} = \begin{cases} 1, & x \leq 0, \\ 0, & x > 0, \end{cases} \quad \text{“≤” switch.} \quad (33)$$

$$S_4(x) = 1 - \frac{1}{1 + e^{(x+\epsilon)\cdot\infty}} = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0, \end{cases} \quad \text{“≥” switch.} \quad (34)$$

Non-constancy of the switches in the proposed rules. The value of x depends on random draws and evolving state/memory variables, so both activation and deactivation are produced:

- In the cardinality rule (Equation (35)), $S_3(\mu_1 - z_j)$ compares a uniform draw $\mu_1 \in [0, 1]$ with cumulative weights $z_j \in [0, 1]$; across j , both cases $x < 0$ and $x > 0$ are obtained, so S_3 toggles. The quantity $\sum_q S_2(w_{i,q}(t))$ counts strictly positive success weights $w_{i,q}(t)$, and S_2 is activated only when $w_{i,q}(t) > 0$.
- In node selection (Equation (36)), $S_3(\mu_2 - z_j)$ operates analogously with a fresh $\mu_2 \in [0, 1]$ and cumulative weights derived from $\mathbf{w}_i(t)$, yielding both $x \leq 0$ and $x > 0$.
- In the clipping function (Equation (39)), $S_3(\beta_j^{(2)} - x)$ and $S_4(\beta_j^{(1)} - x)$ compare the current value x with the moving upper and lower bounds $\beta_j^{(2)}$ and $\beta_j^{(1)}$; depending on the outcome, x is passed through or snapped to the nearest bound.

The ϵ offsets enforce a consistent tie policy: equality is included in $S_3 (\leq)$ and $S_4 (\geq)$ and excluded in S_1 and S_2 . This policy is applied consistently in all the referenced rules.

3.4. Offspring: Phymastichus

This subsection delves into the process governing the generation of new offspring of *P. coffea* agents. As previously discussed, PHA-MD introduces the capability to block nodes, reducing the pool of available nodes. This attribute materially influences the decision-making of each *P. coffea* agent through the memory M . Situations may arise where the number of available nodes is fewer than the destinations desired by an agent. To avoid this condition, the cardinality of each agent $\mathbf{p}_i(t)$ is defined as

$$c_i(t) = \min \left\{ \min_{j \in [1, N_n]} \frac{n_j}{S_3(\mu_1 - z_j)}, \sum S_2(\mathbf{w}_i(t)) \right\},$$

$$\mathbf{z}^{(M)} = [z_1, \dots, z_{N_n}], \quad z_k = \sum_{j=1}^k \frac{m_{i,j}(t)}{\sum \mathbf{m}_i(t)},$$

$$\mathbf{n} = [1, \dots, N_n].$$
(35)

where $S_2(\cdot)$ and $S_3(\cdot)$ are the saturated sigmoid switch functions defined in Section 3.3, and $\mu_1 \in [0, 1]$ is a uniformly distributed random number. Once $c_i(t)$ is selected, the set of pinning nodes is defined (Figure 3) so that the j -th node visited by agent $\mathbf{p}_i(t)$ is

$$p_{i,j}(t) = \min_{j \in [1, c_i(t)]} \frac{n_j}{S_3(\mu_2 - z_j)},$$

$$\mathbf{z}^{(W)} = [z_1, \dots, z_{N_n}], \quad z_k = \sum_{j=1}^k \frac{\omega_j}{\sum \omega},$$

$$\omega = \mathbf{w}_i(t), \quad \mathbf{n} = [1, \dots, N_n].$$
(36)

where $S_3(\cdot)$ is defined in Section 3.3 and $\mu_2 \in [0, 1]$ is uniformly distributed. If node n_j is selected, its corresponding $w_{i,j}(t)$ during iteration t is set to 0 so that n_j is not re-selected by the same agent. The auxiliary vector ω is used for this purpose, allowing temporary modification without altering the original \mathbf{w}_i .

3.5. Offspring: Hypothenemus

In this subsection, the generation process of the offspring in the population $H(t)$ is described. The population $H(t)$ evolves under random perturbations and position increments, and is given by

$$h_{i,j}(t+1) = \begin{cases} \beta_j^{(1)}(0) + (\beta_j^{(2)}(0) - \beta_j^{(1)}(0))\mu_3, & \text{if } t = 1 \vee \tau = 1, \\ o(h_{i,j}(t) + \rho_3 \Delta_{i,j}(t)), & \text{else,} \end{cases}$$
(37)

where $\mu_3 \in [0, 1]$ is uniformly distributed. The position increment of the i -th individual is

$$\Delta_{i,j}(t+1) = \begin{cases} 0, & \text{if } t = 1 \vee \tau = 1, \\ \Delta_{i,j}(t) + \rho_1 \mu_4 (h_{i,j}^{(1)}(t) - h_{i,j}(t)) + \rho_2 \mu_5 (h_{i,j}^{(2)}(t) - h_{i,j}(t)), & \text{else,} \end{cases} \quad (38)$$

where ρ_1 is the local-influence constant (two-female births), ρ_2 is the global-influence constant (female–male births), ρ_3 is an inertial constant, and $(\mu_4, \mu_5) \in [0, 1]^2$ is uniformly distributed.

The clipping function $o(x)$ prevents *H. hampei* agents from exceeding the search space bounds:

$$o(x) = x S_3(\beta_j^{(1)} - x) S_4(\beta_j^{(2)} - x) + S_4(\beta_j^{(1)} - x) \beta_j^{(1)} + S_3(\beta_j^{(2)} - x) \beta_j^{(2)}, \quad (39)$$

where $S_3(\cdot)$ and $S_4(\cdot)$ are the saturated sigmoid switch functions defined in Section 3.3. In this way, values inside the interval $[\beta_j^{(1)}, \beta_j^{(2)}]$ are preserved, whereas values attempting to cross a bound are snapped to the closest limit. This mechanism ensures that the population remains within the feasible search space while adapting to the evolving influence cues.

3.6. Flowchart Analysis of the PHA-MD Algorithm

Figure 6 sketches the overall control flow of PHA-MD. It highlights, at a high level, the cycle that repeats across iterations and epochs: (i) initialization at the start of each epoch; (ii) agent-driven updates—*Phymastichus* constructs the candidate pinning set via the inheritance memories, while *Hypothenemus* refines the continuous gains through the borer and clipping functions; and (iii) the epoch-end blocking step, where visits and successes are evaluated to update the blocking set. The diagram intentionally omits low-level details to emphasize data flow and stage interactions. For completeness, Algorithm 1 provides the corresponding pseudocode, aligned with Section 3, which can be read in parallel with the equations to recover the full procedural detail.

The steps that describe the *Phymastichus* Agent Behavior stage are as follows. “Initialization”: Each agent initializes with a random set of potential pinning. “Node Selection”: Using cumulative probabilities, the agent selects nodes to visit based on the success memories ($W(t)$) and counts ($M(t)$). “Objective Function Evaluation”: The selected nodes are evaluated using the objective function to determine their effectiveness in controlling the network. “Memory Update”: The agent updates its success memories and counts based on the evaluation results (Figure 7).

The steps that describe the *Hypothenemus* Agent Behavior stage are as follows. “Parameter Adjustment”: The agents adjust their parameters ($\beta^{(2)}$ and $\beta^{(1)}$) using the sigmoid functions based on the iteration count. “Delta Update”: Each agent updates its delta values, incorporating local and global influences. “Node Evaluation”: Nodes are evaluated using the clipping function to ensure they remain within the bounds of $\beta^{(1)}$ and $\beta^{(2)}$. “Update Influence”: The agents update the network influence values based on their evaluations (Figure 8).

The steps that describe the Update and Evaluation of the Blocking Group stage are as follows. “Epoch Check”: The algorithm checks if the current epoch or iteration limit has been reached. “Node Blocking”: Nodes are blocked based on their visitation and success rates. “Reset Counters”: Visitation and success memories are reset for the next iteration. “Stability Check”: The algorithm checks the stability of the network and updates the best blocking group if stability is achieved. In summary, these flowcharts provide a visual representation of the PHA-MD algorithm’s structure and operations, offering insights into the systematic and iterative processes that underlie its effectiveness in optimizing pinning control for complex networks (Figure 9).

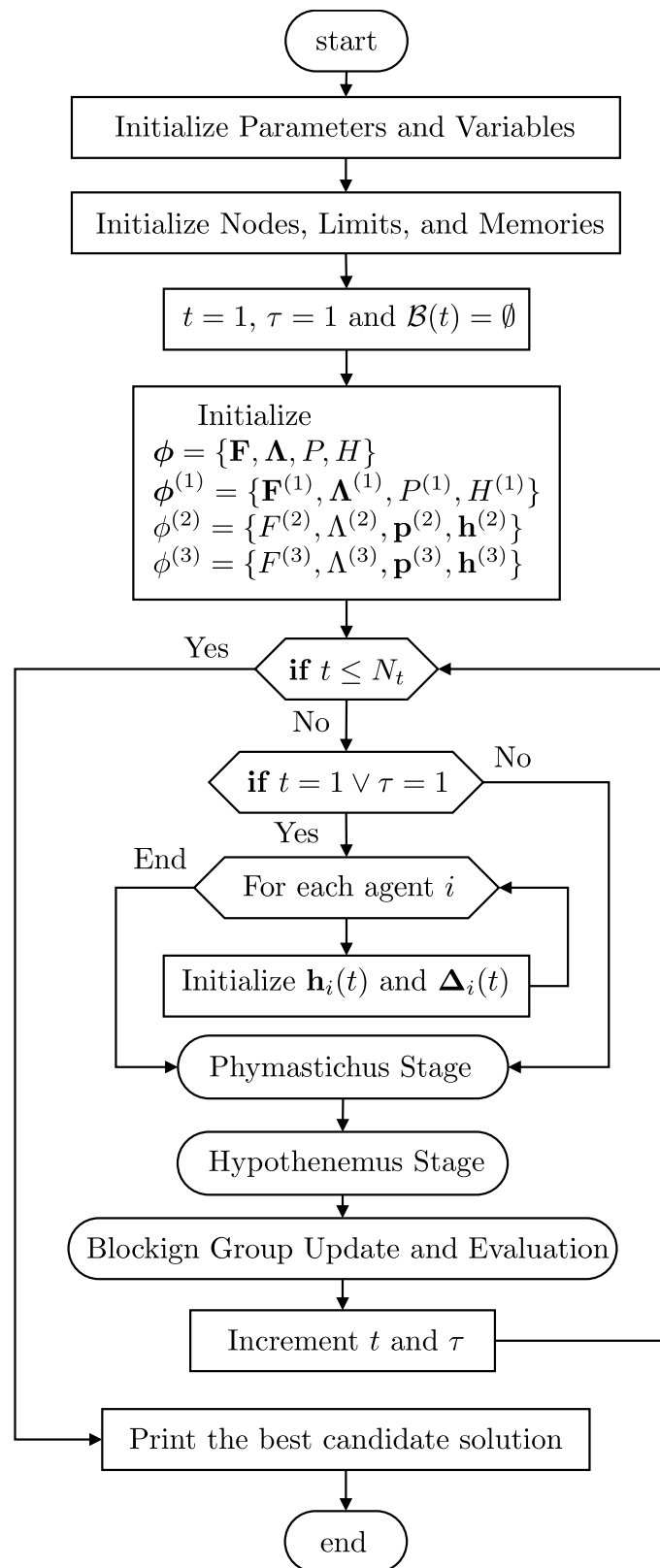


Figure 6. General loop.

Algorithm 1 PHA-MD

Data: $N_n, N_t, N_\tau, N_{\lambda^+}, N_a$, bounds $\beta^{(1)}(0) = \mathbf{l}_0, \beta^{(2)}(0) = \mathbf{u}_0, \zeta, (\rho_1, \rho_2, \rho_3)$

Result: $\phi^{(2)}$ (global best), $\phi^{(3)}$ (best under blocking), $\mathcal{B}, \mathcal{B}_{\max}$

```

1: Init:  $t \leftarrow 1, \tau \leftarrow 1, \mathcal{B} \leftarrow \emptyset, \mathcal{B}_{\max} \leftarrow \emptyset$ 
2:  $W \leftarrow \mathbf{1}_{N_a \times N_n}, M \leftarrow \mathbf{1}_{N_a \times N_n}$  with  $M(:, 1:N_{\lambda^+}) \leftarrow 0, V \leftarrow \mathbf{0}$ 
3: Records:  $\phi$  (current),  $\phi^{(1)}$  (per-agent best),  $\phi^{(2)}$  (global best),  $\phi^{(3)}$  (blocked best)
4: while  $t \leq N_t$  do
5:   if  $t = 1$  or  $\tau = 1$  then ▷ epoch start
6:     Initialize  $H$  within  $[\mathbf{l}_0, \mathbf{u}_0]$  as in Equation (37); set  $\Delta \leftarrow \mathbf{0}$ 
7:   end if
8:   for  $i = 1$  to  $N_a$  do ▷ Phymastichus: node selection via memories
9:     Compute  $Z_M = \text{cumsum}(M(i, :) / \sum M(i, :))$ 
10:     $c_i \leftarrow \text{CARDINALITYBY}$  Equation (35) using  $Z_M, S_2, S_3$ 
11:     $\omega \leftarrow W(i, :)$  ▷ temporary copy to avoid reselection
12:     $\mathbf{p}_i \leftarrow \emptyset$ 
13:    for  $j = 1$  to  $c_i$  do
14:       $Z_W = \text{cumsum}(\omega / \sum \omega)$ 
15:      Select  $p \leftarrow \text{SELECTNODEBY}$  Equation (36) using  $Z_W, S_3$ ; set  $\omega(p) \leftarrow 0$  ▷ no reselection
16:       $\mathbf{p}_i \leftarrow \mathbf{p}_i \cup \{p\}; V(i, p) \leftarrow V(i, p) + 1$ 
17:    end for
18:    Evaluate  $F_i, h_i$  by Equation (15); form  $\phi_i(t)$  (Equation (13))
19:    Update  $\phi_i^{(1)}$  by Equation (23); if improved then  $W(i, \mathbf{p}_i) += 1, M(i, |\mathbf{p}_i|) += 1$ 
20:    if  $h_i = 0$  then ▷ stable at current size
21:       $M(:, |\mathbf{p}_i| + 1 : N_n) \leftarrow 0$  ▷ prune larger cardinalities
22:    end if
23:  end for
24:  Update  $\phi^{(2)}$  by Equation (24) using  $F_{\min}^{(1)}(t)$  and argmin  $k$ 
25:  Broadcast  $\phi^{(2)}.H \leftarrow (\phi^{(1)}.H \text{ at row } k) \text{ replicated to } N_a \text{ rows}$ 
26:  if  $\phi^{(2)}$  improved this step then ▷ as in code: only if  $f_{\min}$  improves
27:    Draw random subset  $R \subset \{1, \dots, N_a\}$ ; set  $W(R, \phi^{(2)}.p) \leftarrow W(R, \phi^{(2)}.p) + 1$ 
28:  end if
29:  Hypothenemus: borer & clipping
30:  Update lower/upper bounds  $\beta^{(1)}, \beta^{(2)}$  by Equations (21) (lower), (22) (upper) using  $\phi^{(2)}.H, \zeta, \mathbf{l}_0, \mathbf{u}_0$ 
31:   $\Delta \leftarrow \Delta + \rho_1 \text{rand} \circ (\phi^{(1)}.H - H) + \rho_2 \text{rand} \circ (\phi^{(2)}.H - H)$  ▷ Equation (38)
32:   $H \leftarrow o(H + \rho_3 \Delta, \beta^{(1)}, \beta^{(2)})$  ▷ Equation (39) with  $S_3, S_4$ 
33:  if  $\tau = N_\tau$  or  $t = N_t$  then ▷ blocking stage
34:     $b \leftarrow \arg \max_i \sum_j \frac{V(j, i)}{W(j, i)}$  ▷ Equation (28)
35:    Reset epoch counters:  $V \leftarrow \mathbf{0}, W \leftarrow \mathbf{1}, M \leftarrow \mathbf{0}, \tau \leftarrow 0$ 
36:    if  $\phi^{(2)}.h = 0$  then ▷ stability at epoch end
37:      if  $\mathcal{B} = \emptyset$  then
38:         $\phi^{(3)} \leftarrow \phi^{(2)}; \mathcal{B} \leftarrow \{b\}; \mathcal{B}_{\max} \leftarrow \mathcal{B}$ 
39:      else
40:        If  $|\phi^{(2)}.p| < |\phi^{(3)}.p|$  or (tie &  $F^{(2)} < F^{(3)}$ ):  $\phi^{(3)} \leftarrow \phi^{(2)}$ 
41:        If  $|\mathcal{B}| \geq |\mathcal{B}_{\max}|$ :  $\mathcal{B}_{\max} \leftarrow \mathcal{B}$ 
42:        if  $N_n - |\mathcal{B}| > N_{\lambda^+}$  then
43:           $\mathcal{B} \leftarrow \text{unique}(\mathcal{B} \cup \{b\})$ 
44:        end if
45:      end if
46:       $W(:, \mathcal{B}) \leftarrow \mathbf{0}; M(:, N_n - |\mathcal{B}| - 1 : N_n - |\mathcal{B}|) \leftarrow \mathbf{1}$ 
47:      else if  $\mathcal{B} \neq \emptyset$  and  $\phi^{(2)}.h \neq 0$  then ▷ recover best blocking if no stability
48:         $\mathcal{B} \leftarrow \mathcal{B}_{\max}; W(:, \mathcal{B}) \leftarrow \mathbf{0}; M(:, N_n - |\mathcal{B}| - 1 : N_n - |\mathcal{B}|) \leftarrow \mathbf{1}$ 
49:      end if
50:    end if
51:     $t \leftarrow t + 1; \tau \leftarrow \tau + 1$ 
52:  end while
53: return  $\phi^{(2)}, \phi^{(3)}, \mathcal{B}, \mathcal{B}_{\max}$ 

```

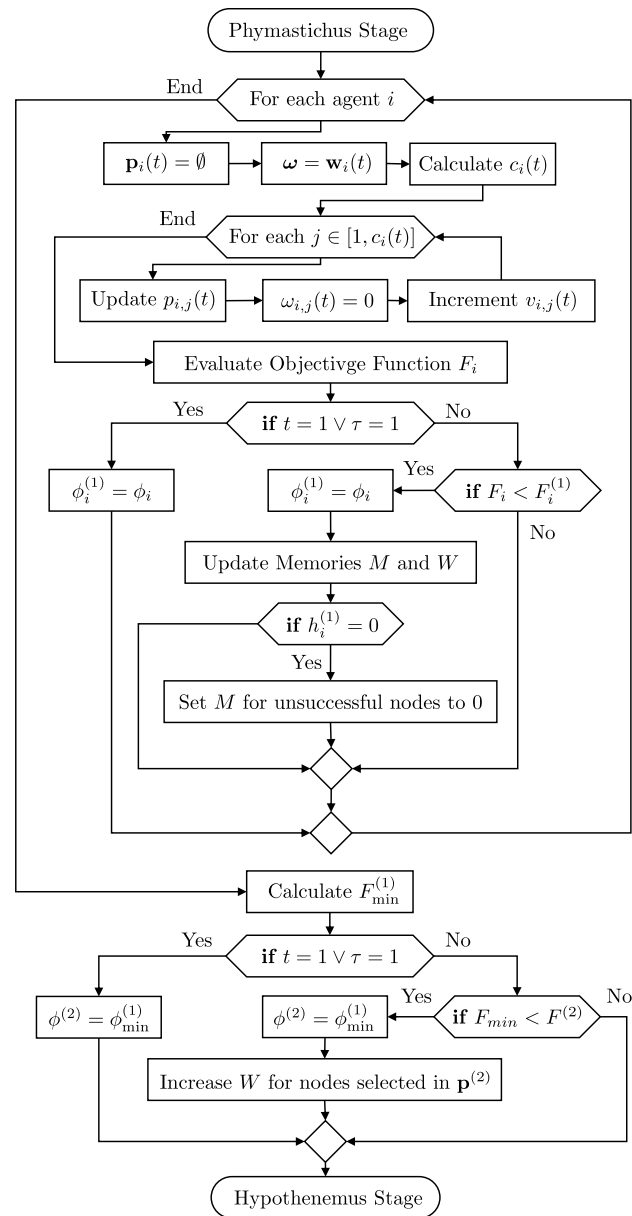


Figure 7. Phymastichus agent behavior.

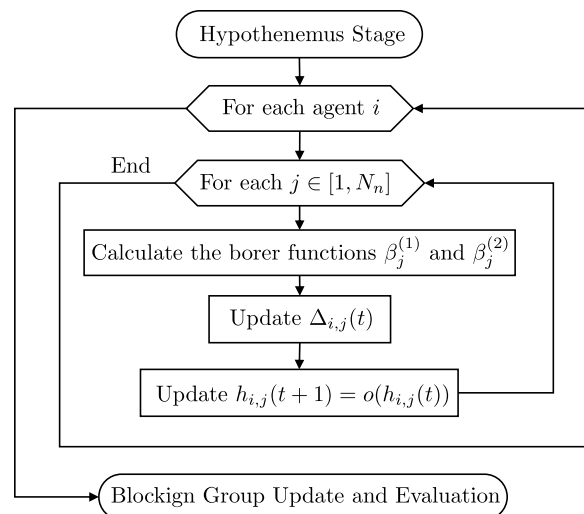


Figure 8. Hypothenemus agent behavior.

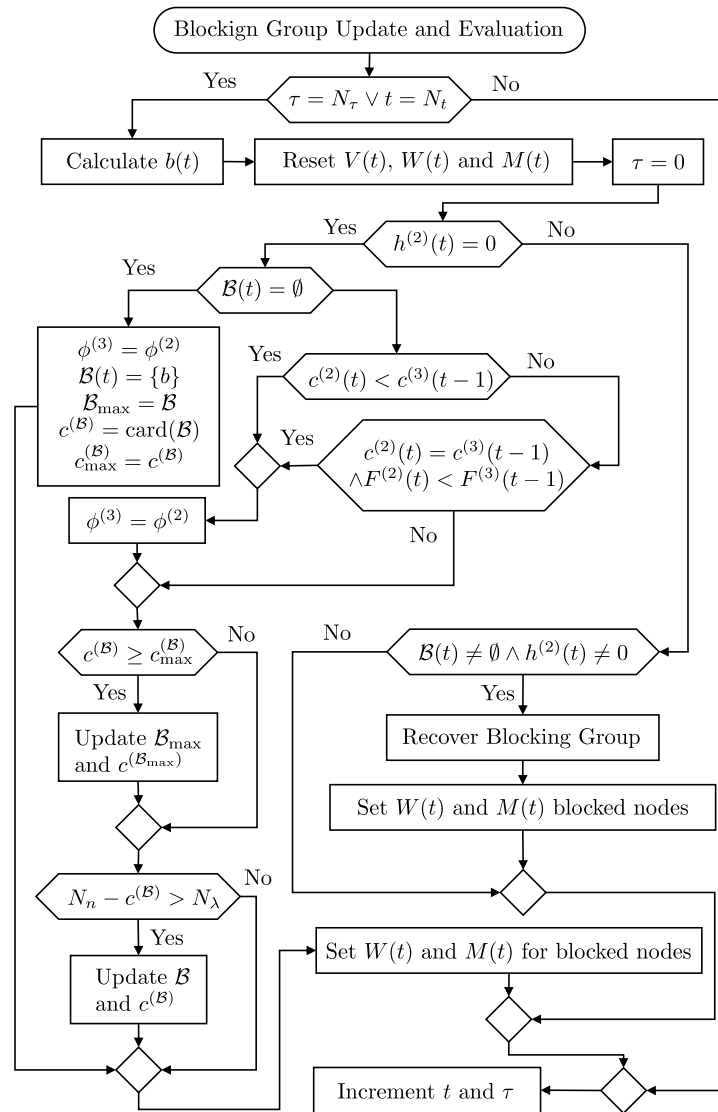


Figure 9. Updating and evaluation of the blocking group.

3.7. Spatial Complexity Analysis

The spatial complexity of the proposed algorithm is determined by analyzing the memory usage in relation to the input parameters, specifically the number of agents N_a and the number of nodes N_n in the complex network. The key data structures contributing to memory consumption include matrices and cell arrays that are dynamically allocated during the algorithm's execution.

The primary matrices W , M , V , and ϕ , as well as the matrices $\beta^{(1)}$ and $\beta^{(2)}$, each have a size of $N_a \times N_n$. Each of these matrices contributes $O(N_a \times N_n)$ to the spatial complexity. Since there are six such matrices, their combined complexity is $O(6 \times N_a \times N_n)$.

The cell array \mathbf{p} , where each cell contains a vector of selected nodes, also contributes to the spatial complexity. In the worst case, the memory required is $O(N_a \times N_n)$. Additional vectors, such as F , h , and their counterparts in $\phi^{(1)}$, $\phi^{(2)}$, and $\phi^{(3)}$, have a size of N_a , contributing $O(6 \times N_a)$ in total. When combined, the total spatial complexity is

$$S(n) = O(6 \times N_a \times N_n) + O(N_a \times N_n) + O(6 \times N_a). \quad (40)$$

Given that the term $O(N_a \times N_n)$ dominates, the spatial complexity can be simplified to $S(n) = O(N_a \times N_n)$. This indicates that the memory usage of the algorithm increases

linearly with both the number of agents and the number of nodes in the complex network, demonstrating the algorithm's spatial efficiency.

4. Simulation Study

It is important to note that the proposed PHA-MD algorithm features variable dimensions across its agents, with each agent having a different number of dimensions. This characteristic initially precluded a direct comparison with other algorithms, such as ALO [21], TLBO [22], GWO [23], AMO [24], PSO [25], ABC [26], GSK [27], BBO [28], WOA [29], ACO [30], OOA [31], MA [32], AOA [33], CHIO [34], and DTBO [35]. These algorithms assume a fixed number of dimensions throughout the optimization process, which is not compatible with the dynamic dimensionality of PHA-MD.

The optimization problem addressed by PHA-MD involves both combinatorial and continuous elements, further complicating the use of traditional algorithms. PHA-MD was specifically designed to handle the dynamic dimensionality needed to identify the minimum number of pinning nodes, a task that requires agents to propose different sets of nodes. This flexibility in dimension handling is not supported by the aforementioned algorithms, which rely on a fixed dimension setup. For instance, initializing agents with a predefined number of dimensions for both gain constants and potential pinning nodes imposes assumptions that may not be accurate.

To allow a fair comparison, a reconditioning function was introduced. This function adjusts the dimensions by rounding them to the nearest integer corresponding to node indices in the matrix K and selecting unique elements. This approach standardizes the input, enabling fixed-dimension algorithms to adapt to the variable dimensionality required by PHA-MD.

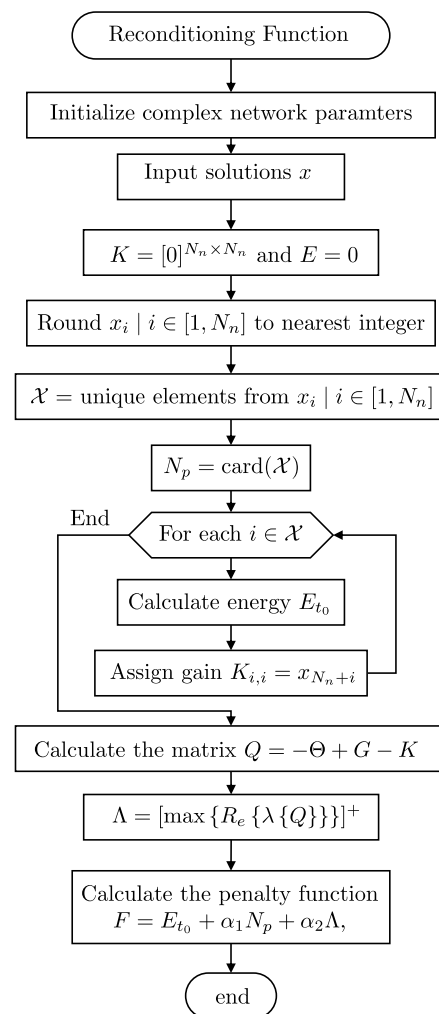
All runs were performed on a single workstation under Ubuntu 22.04.3 LTS (64-bit) using MATLAB R2023b in double precision. Default multithreaded BLAS (Intel oneMKL) was enabled; GPU acceleration was not used. Randomness was controlled with MATLAB's `rng('twister', 2024)`. The population size, iteration budget, and epoch length followed the values stated in Section 3, PHA-MD Algorithm, and the figure set *Convergence curves for the Networks*. The hardware and software stack is summarized in Table 1.

Remark 2. *The number of runs required to obtain a representative solution cannot be specified a priori in a universal manner, since the selection of pinning nodes constitutes a combinatorial NP-hard problem. Consequently, the evaluation protocol is grounded in a theoretical feasibility criterion: under the V-stability of Xiang and Chen, the number of pinned nodes cannot be less than the number of non-negative eigenvalues of the characteristic matrix; thus, $N_{\lambda+}$ is a natural lower bound. Operationally, each method was executed up to the point at which the baseline algorithms exhibited instability, whereas PHA-MD preserved feasible solutions. Accordingly, a finite iteration budget was adopted as the stopping criterion; its value was selected experimentally to reach a regime in which baseline methods typically displayed instability in their solutions, while PHA-MD continued to yield feasible trajectories. Stability therefore informs the choice of iteration budget, although simulations are not terminated upon the detection of instability itself.*

Figure 10 illustrates the reconditioning function used in the simulation study. This function is essential for aligning fixed-dimension algorithms with the variable dimensionality necessary for pinning node selection in complex networks. The properties of the six complex networks analyzed are presented in Table 2. These properties include the number of edges (N_e), positive eigenvalues ($N_{\lambda+}$), average connection degree (\overline{cd}), average coupling strength (\overline{c}), average passivity degree ($\overline{\theta}$), average initial conditions ($\overline{x_0}$), and their respective distribution intervals (I_{c,θ,x_0}). Additionally, the search space (SS) defines the feasible region for selecting control gains during the optimization process.

Table 1. Hardware and software used for all computations.

Component	Specification
CPU	Intel Core i7-12700K (12 cores: 8P+4E, 20 threads, 3.6–5.0 GHz)
Memory	32 GB DDR4-3200
GPU	NVIDIA GeForce RTX 3060, 12 GB VRAM (not used)
Storage	1 TB NVMe SSD
Operating system	Ubuntu 22.04.3 LTS (64-bit), Linux kernel 5.15
MATLAB	R2023b (double precision)
Toolboxes	Optimization, Global Optimization; Parallel Computing (installed, disabled)
Math kernel	Intel oneMKL (default in MATLAB R2023b)
Threading	Default multithreaded linear algebra; no GPU acceleration
Random seed	<i>rng('twister', 2024)</i>
Measurement	MATLAB <i>tic/toc</i> ; single-process execution

**Figure 10.** A flowchart of the reconditioning function.

The six synthetic networks were sampled uniformly within the intervals reported in Table 2 under two design criteria: (i) passivity degrees θ and coupling strengths c_{ij} chosen to yield a nonzero count of positive eigenvalues N_{λ^+} in the V-stability analysis, and (ii) a relatively large N_{λ^+} compared to N_n , which increases the theoretical lower bound on the number of required pinning nodes. This produces challenging instances where stability is non-trivial and energy–cardinality trade-offs are visible. Although synthetic, these topologies are representative of interaction patterns commonly found in signed social influence graphs (trust/distrust) [45], neuronal microcircuits with mixed excitatory/inhibitory synapses [46],

gene-regulatory subnetworks with activation/repression [47], distribution grids with grid-connected power-electronic converters exhibiting impedance interactions [48,49], and antagonistic ecological webs (predator–prey) [50]. Thus, “Network 1–6” should be read as topology classes that capture these motifs, rather than as single specific datasets.

Table 2. Complex network data analyzed.

	Network 1	Network 2	Network 3	Network 4	Network 5	Network 6
N_n	10	20	30	40	50	60
N_e	18	93	224	383	653	876
N_{λ^+}	3	7	5	21	25	33
\overline{cd}	3.6	9.3	14.93	19.15	26.12	29.2
$\min\{cd\}$	2	5	10	12	19	19
$\max\{cd\}$	7	13	20	24	34	38
\bar{c}	10.92	5.78	9.53	−11.35	−6.90	−4.64
I_c	[−4, 30]	[−2, 15]	[−2, 20]	[−40, 20]	[−24, 10]	[−24, 16]
$\bar{\theta}$	6.69	6.31	45.93	−6.31	0.46	−7.69
I_θ	[−1, 13]	[−10, 15]	[−4, 80]	[−23, 8]	[−16, 12]	[−56, 35]
\bar{x}_0	$\begin{bmatrix} 16.85 \\ -10.73 \\ -10.2 \end{bmatrix}$	$\begin{bmatrix} -22.13 \\ 1.03 \\ -12.59 \end{bmatrix}$	$\begin{bmatrix} -0.01 \\ 2.02 \end{bmatrix}$	$\begin{bmatrix} 2.02 \\ 2.21 \end{bmatrix}$	$\begin{bmatrix} -2.33 \\ -2.83 \end{bmatrix}$	$\begin{bmatrix} 23.83 \\ 26 \\ 31.85 \end{bmatrix}$
I_{x_0}	[−80, 70]	[−90, 55]	[−8, 9]	[−2, 7]	[−14, 8]	[−23, 80]
SS	$\begin{bmatrix} 10 \\ 1 \times 10^{10} \end{bmatrix}$	$\begin{bmatrix} 1 \times 10^4 \\ 2 \times 10^5 \end{bmatrix}$	$\begin{bmatrix} 1 \times 10^2 \\ 1 \times 10^7 \end{bmatrix}$	$\begin{bmatrix} 1 \times 10^4 \\ 1 \times 10^6 \end{bmatrix}$	$\begin{bmatrix} 1 \times 10^5 \\ 1 \times 10^7 \end{bmatrix}$	$\begin{bmatrix} 1 \times 10^7 \\ 1 \times 10^9 \end{bmatrix}$

The parameter settings used in the simulations for the various algorithms are summarized in Table 3. For each baseline, hyperparameters were taken from the authors’ recommended operating conditions in their original sources; no per-benchmark retuning was performed to avoid overfitting to the synthetic networks. The population size and number of iterations were kept constant across all methods to enforce a common computational budget. A reconditioning wrapper was employed to standardize candidate representations so fixed-dimension heuristics could accept the variable-cardinality solutions produced by PHA–MD. Because the control objective was stability-first, candidates that failed the stability test are reported as unstable and are not considered acceptable low-energy optima. This protocol separates algorithmic design differences (fixed versus variable dimensionality with stability feedback) from hyperparameter tweaks and supports a fair comparison.

As mentioned, the performance of the proposed PHA–MD algorithm is compared with several other heuristic optimization algorithms across six different complex network topologies. These topologies were generated uniformly at random within the intervals specified in Table 2. The networks were designed to challenge current optimization algorithms and reveal scenarios where stability is difficult to achieve. The aim was to create increasingly complex networks to showcase the superior capability of the PHA–MD algorithm. If a network were encountered where even PHA–MD could not find stable solutions, the algorithm would need to be reconfigured to address these new challenges.

The simulation results include two sets of graphs. The first set (Figures 11–13) presents the convergence curves for each network’s optimization, comparing the performance of the PHA–MD algorithm with that of other algorithms. The second set (Figures 14–16) of graphs shows the optimization of energy E_{t_0} against the percentage of unstable solutions, emphasizing solution stability.

Table 3. The parameter settings for the algorithms used in the simulations.

Algorithm	Parameter	Value
ALO	Elite rate	0.1
	Ant rate	0.9
WOA	Convergence parameter (a)	[2, 0]
	r (random vector)	[0, 1]
	l (random number)	[−1, 1]
TLBO	T_F (teaching factor)	[(1 + rand)]
	rand (random number)	[0, 1]
GWO	a (convergence parameter)	[2, 0]
	α (alternative parameter)	$2 - 2(g/\max_g)$
AMO	Animals in each group	5
PSO	ω (inertia weight)	0.6
	c_1, c_2 (cognitive and social constants)	2
ABC	Abandonment criteria	25
GSK	P (probability)	0.1
	k_f (forward coeff)	0.5
	k_r (return coeff)	0.9
	K (number of iters)	10
BBO	Habitat modification probability	1
	Immigration probability bounds per gene	[0, 1]
	Step size for numerical integration	1
	Max. immigration and migration rates	1
	Mutation probability	0.1
ACO	Initial pheromone value	1×10^{-6}
	Pheromone update constant	20
	Exploration constant	1
	Global pheromone decay rate	0.9
	Local pheromone decay rate	0.5
	Pheromone sensitivity	1
	Visibility sensitivity	5
PHA	ρ_1 (local influence coeff)	0.5
	ρ_2 (global influence coeff)	0.6
	ρ_3 (increment coeff)	0.7
	ζ (clustering tolerance)	0.02
	γ (control gain)	10
OOA	Elite rate	0.2
	Infection rate	0.1
MA	Attraction rate	0.5
	Repulsion rate	0.3
AOA	Exploration rate	0.4
	Exploitation rate	0.6
CHIO	Infection probability	0.3
	Recovery probability	0.1
DTBO	Learning rate	0.1
	Feedback rate	0.2
All algorithms	Population size	50
	Number of iters	500

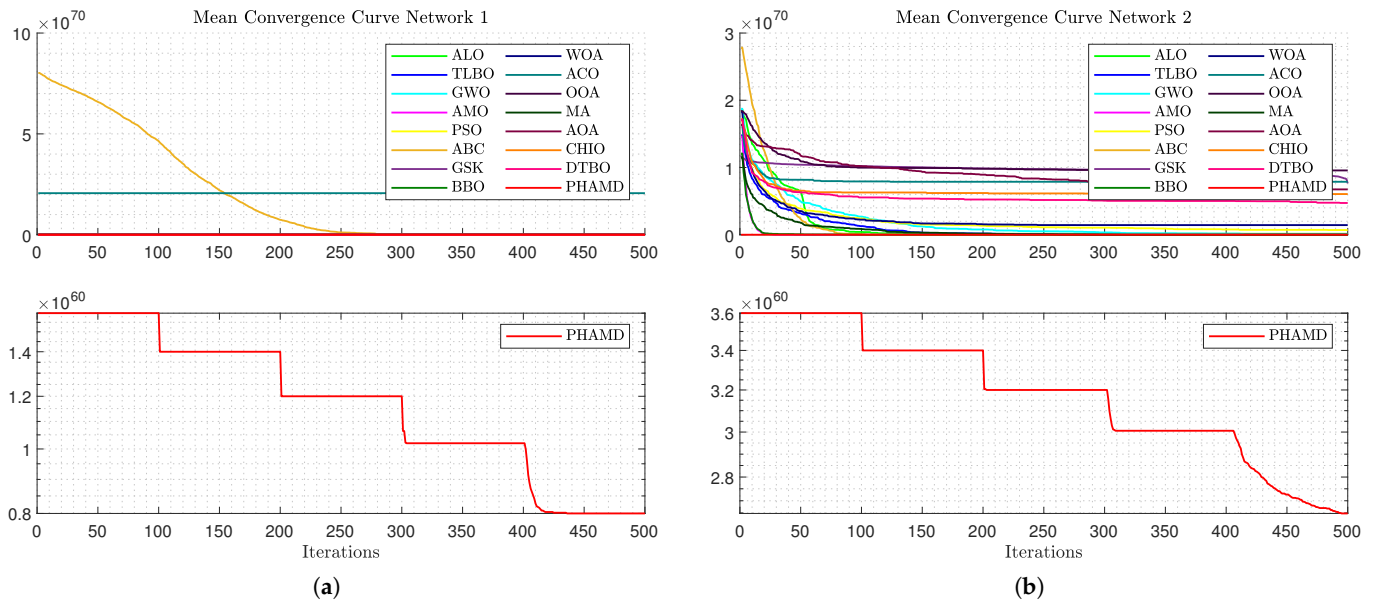


Figure 11. Convergence curves for Networks 1 and 2.

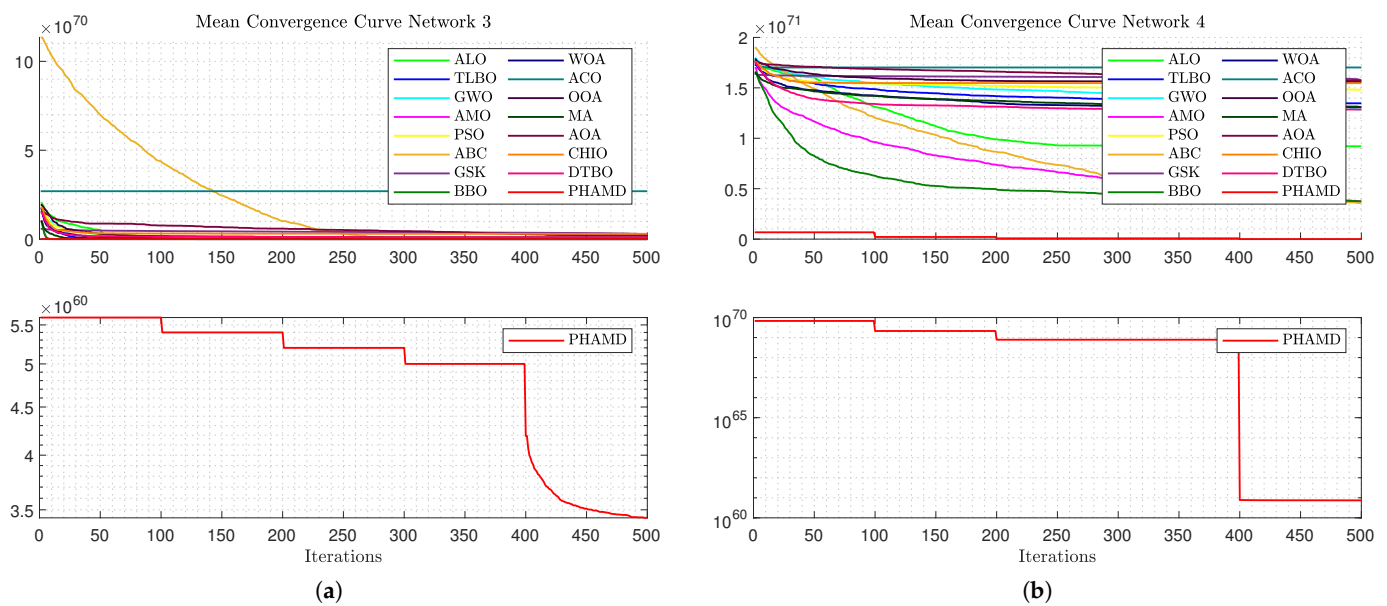


Figure 12. Convergence curves for Networks 3 and 4.

In Figure 11a, the convergence behavior for Network 1 is depicted. Due to the scale, only the ABC algorithm is prominently visible, with others appearing unchanged. Figure 11b allows for a comparison across all algorithms, showing similar convergence except for in the case of GSK and ACO.

The PHA-MD algorithm reaches its minimum value earlier than others, demonstrating its efficiency. Figure 12a complicates comparisons due to the ABC algorithm's scale, yet most algorithms show early convergence, with PHA-MD continuing to converge rapidly.

As shown in Figure 12b, the optimization process becomes more challenging due to Network 4's size, with a clear gap between PHA-MD and the others, and only four out of five epochs observed due to unchanged candidate solutions. Figure 13a displays similar behavior, with drilling functions proving effective, and BBO and AMO following PHA-MD in performance. Finally, Figure 13b shows an even greater gap between PHA-MD and the

other algorithms, highlighting PHA-MD's superior performance in minimizing the cost function F .

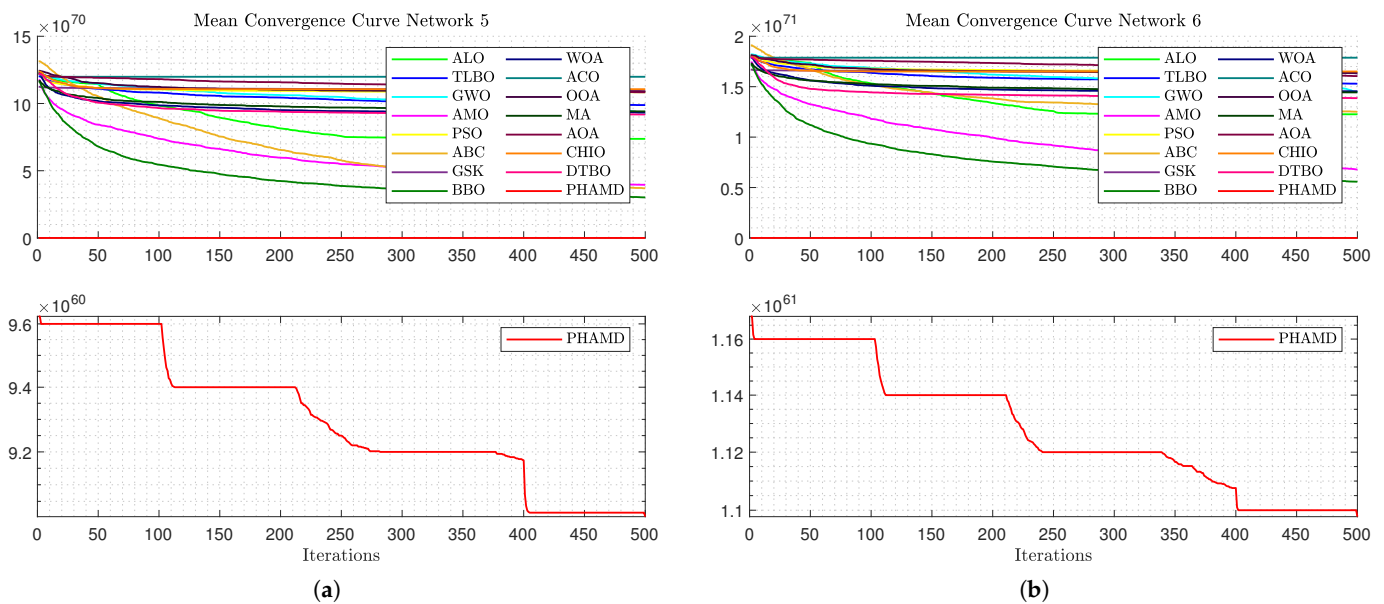


Figure 13. Convergence curves for Networks 5 and 6.

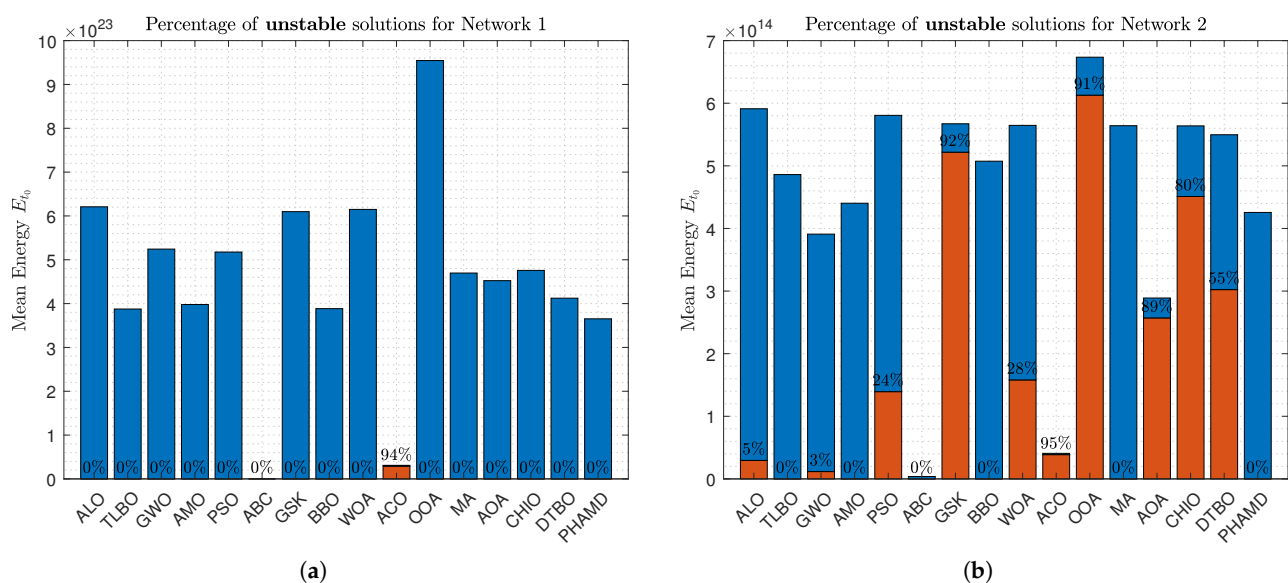


Figure 14. Energy optimization and percentage of unstable solutions for Networks 1 and 2.

In addition to the convergence curves, Figures 14–16 present the optimization of energy E_{t_0} against the percentage of unstable solutions. Blue bars represent the average energy consumption of each algorithm's solutions, with percentages indicating the proportion of unstable solutions. Figure 14a shows that the ABC algorithm achieves the lowest energy consumption with 0% unstable solutions, while ACO, despite being energy-efficient, has 94% unstable solutions, highlighting the trade-off between energy efficiency and stability.

In Figure 14b, the ABC algorithm remains the most energy-efficient, but several algorithms, including ALO, GWO, PSO, WOA, GSK, and ACO, present unstable solutions, with ACO having the highest instability rate at 95%. Figure 15a shows a reduction in unstable solutions due to Network 3 having fewer states. By Figure 15b, nearly all algorithms exhibit 100% unstable solutions, despite Network 4 having the same number of states, indi-

cating that complexity depends on more than just the number of states. Figure 16a shows some algorithms achieving stable solutions, while Figure 16b highlights that only PHA-MD provides stable solutions, underscoring its prioritization of stability over energy efficiency.

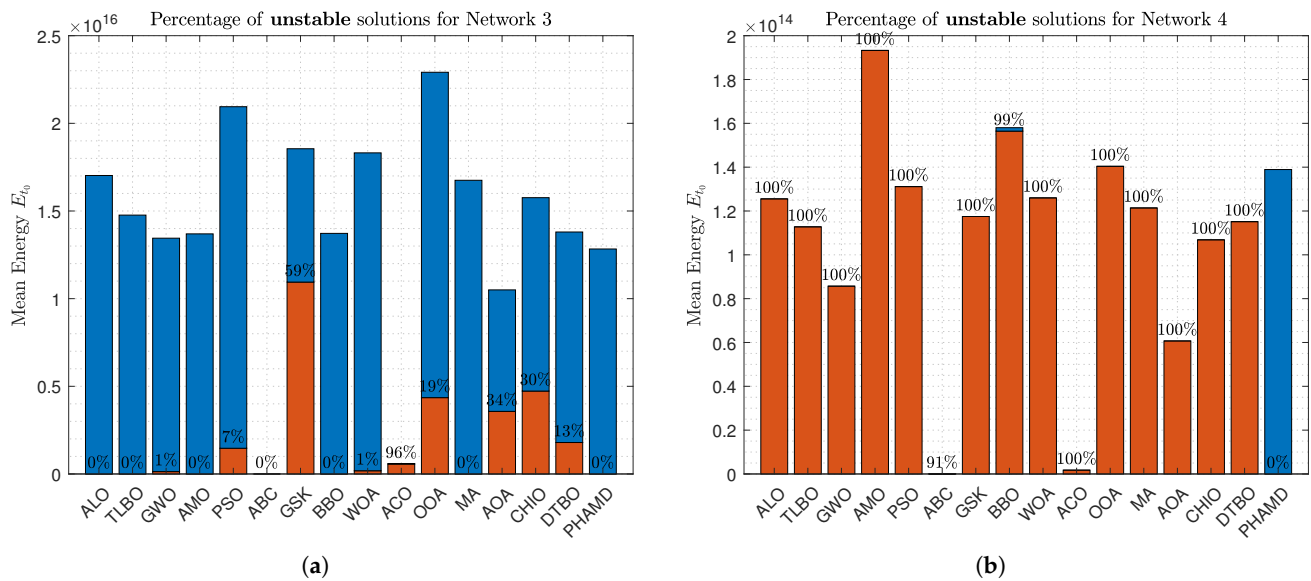


Figure 15. Energy optimization and percentage of unstable solutions for Networks 3 and 4.

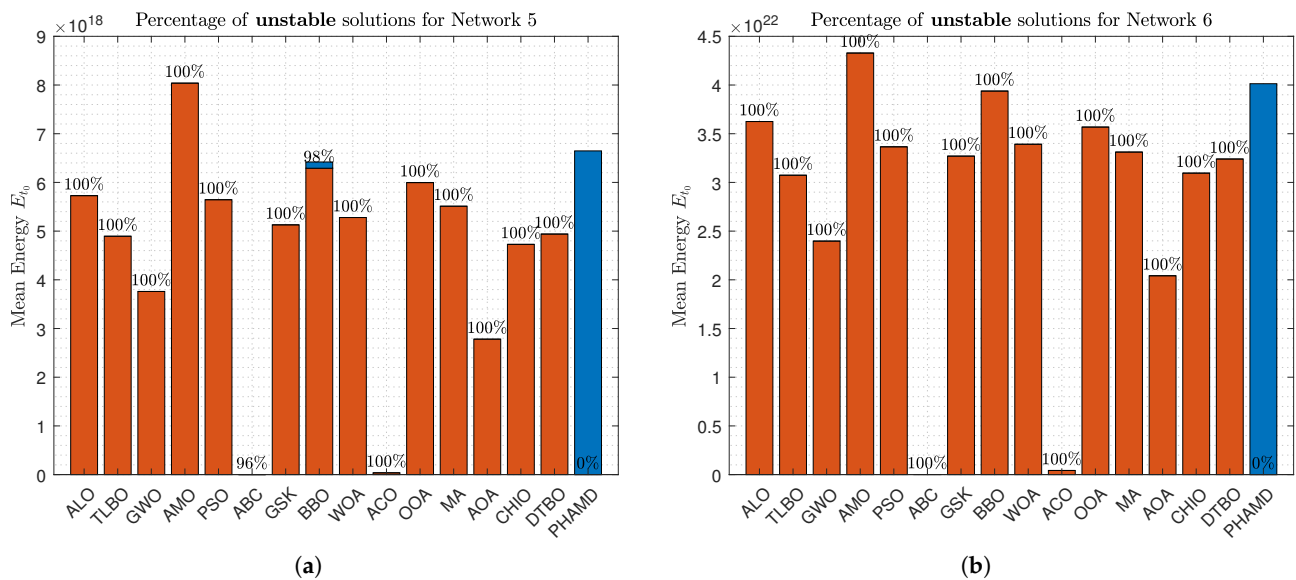


Figure 16. Energy optimization and percentage of unstable solutions for Networks 5 and 6.

These results demonstrate the PHA-MD algorithm's effectiveness in achieving rapid convergence and superior optimization compared to other algorithms. Its ability to handle variable dimensionality and combinatorial optimization makes it particularly suitable for pinning control in complex networks. The simulation results are summarized in Table 4, which provides a comprehensive overview of each algorithm's performance across the six networks.

Table 4. Simulation results for different algorithms across six networks.

Algorithm	Measure	Network 1	Network 2	Network 3	Network 4	Network 5	Network 6
ALO	Avg. CF	5.1×10^{60}	6.8602×10^{68}	1.912×10^{61}	4.6041×10^{71}	3.6848×10^{71}	6.132×10^{71}
	Best CF	4×10^{60}	1.3×10^{61}	1.7×10^{61}	1.7079×10^{71}	1.6174×10^{71}	3.2368×10^{71}
	Avg. NP	5	15	19	33	40	46
	Best NP	4	13	17	30	35	40
	Avg. EC	6.2082×10^{23}	5.9117×10^{14}	1.7025×10^{16}	1.2553×10^{14}	5.7283×10^{18}	3.6252×10^{22}
	Best EC	1.1201×10^{23}	2.404×10^{14}	7.8747×10^{15}	3.7911×10^{13}	2.5242×10^{18}	2.3748×10^{22}
TLBO	Avg. CF	4×10^{60}	1.498×10^{61}	1.962×10^{61}	6.7339×10^{71}	4.942×10^{71}	7.6473×10^{71}
	Best CF	4×10^{60}	1.4×10^{61}	1.7×10^{61}	5.7775×10^{71}	3.6628×10^{71}	5.9984×10^{71}
	Avg. NP	4	15	20	31	37	43
	Best NP	4	14	17	27	33	37
	Avg. E	3.8762×10^{23}	4.8607×10^{14}	1.4766×10^{16}	1.1278×10^{14}	4.8949×10^{18}	3.0739×10^{22}
	Best E	4.5673×10^{22}	1.9126×10^{14}	8.0718×10^{15}	5.7146×10^{13}	2.0877×10^{18}	1.382×10^{22}
GWO	Avg. CF	4.75×10^{60}	2.4728×10^{68}	1.6844×10^{67}	6.434×10^{71}	4.6174×10^{71}	7.2403×10^{71}
	Best CF	4×10^{60}	1.2×10^{61}	1.8×10^{61}	3.745×10^{71}	2.6409×10^{71}	3.9817×10^{71}
	Avg. NP	5	15	20	30	37	43
	Best NP	4	12	18	26	33	37
	Avg. E	5.2435×10^{23}	3.9094×10^{14}	1.3449×10^{16}	8.5704×10^{13}	3.7608×10^{18}	2.3976×10^{22}
	Best E	8.0848×10^{22}	1.1117×10^{14}	2.5247×10^{15}	1.9739×10^{13}	1.0436×10^{18}	1.268×10^{22}
AMO	Avg. CF	4×10^{60}	1.282×10^{61}	1.653×10^{61}	1.8614×10^{71}	1.9714×10^{71}	3.3892×10^{71}
	Best CF	4×10^{60}	1.2×10^{61}	1.5×10^{61}	2.5401×10^{70}	1.624×10^{70}	2.0201×10^{71}
	Avg. NP	4	13	17	36	44	51
	Best NP	4	12	15	34	41	46
	Avg. E	3.979×10^{23}	4.4029×10^{14}	1.3695×10^{16}	1.9327×10^{14}	8.0387×10^{18}	4.3284×10^{22}
	Best E	5.5216×10^{22}	1.9064×10^{14}	7.1131×10^{15}	1.0368×10^{14}	5.0968×10^{18}	2.4297×10^{22}
PSO	Avg. CF	4.79×10^{60}	3.4255×10^{69}	1.4353×10^{69}	7.4066×10^{71}	5.4168×10^{71}	$8.231e+71$
	Best CF	4×10^{60}	1.3×10^{61}	2×10^{61}	6.2805×10^{71}	4.7809×10^{71}	6.5008×10^{71}
	Avg. NP	5	15	22	29	35	41
	Best NP	4	13	20	24	31	35
	Avg. E	5.1762×10^{23}	5.8063×10^{14}	2.0949×10^{16}	1.3113×10^{14}	5.6448×10^{18}	3.3657×10^{22}
	Best E	4.7222×10^{22}	2.3678×10^{14}	8.6401×10^{15}	5.8626×10^{13}	3.1002×10^{18}	1.9188×10^{22}
ABC	Avg. CF	7.84×10^{60}	1.496×10^{61}	2.659×10^{61}	1.7899×10^{71}	1.8478×10^{71}	6.2455×10^{71}
	Best CF	6×10^{60}	1.3×10^{61}	2.4×10^{61}	3.7×10^{61}	4.7×10^{61}	3.7078×10^{71}
	Avg. NP	8	15	27	37	45	48
	Best NP	6	13	24	32	39	41
	Avg. E	6.0857×10^{13}	3.8347×10^{12}	2.0035×10^7	4.4134×10^{10}	1.9193×10^{13}	1.0839×10^{19}
	Best E	6.3576×10^7	3.3042×10^{12}	1.0895×10^7	3.7038×10^{10}	1.4898×10^{13}	8.7471×10^{18}
GSK	Avg. CF	5.7×10^{60}	4.7397×10^{70}	2.0125×10^{70}	7.9435×10^{71}	5.601×10^{71}	8.3242×10^{71}
	Best CF	4×10^{60}	1.6×10^{61}	1.9×10^{61}	6.2524×10^{71}	4.7282×10^{71}	7.0532×10^{71}
	Avg. NP	6	15	22	29	36	42
	Best NP	4	13	19	25	32	36
	Avg. E	6.0977×10^{23}	5.672×10^{14}	1.8551×10^{16}	1.1749×10^{14}	5.1288×10^{18}	3.27×10^{22}
	Best E	5.7557×10^{22}	1.4781×10^{14}	7.3913×10^{15}	5.3564×10^{13}	1.9935×10^{18}	2.2152×10^{22}
BBO	Avg. CF	4.27×10^{60}	1.441×10^{61}	1.733×10^{61}	1.8776×10^{71}	1.5097×10^{71}	2.7855×10^{71}
	Best CF	4×10^{60}	1.3×10^{61}	1.5×10^{61}	3.9×10^{61}	4.5×10^{61}	1.2984×10^{70}
	Avg. NP	4	14	17	36	45	52
	Best NP	4	13	15	31	41	45
	Avg. E	3.8848×10^{23}	5.0742×10^{14}	1.3723×10^{16}	1.5802×10^{14}	6.4197×10^{18}	3.9379×10^{22}
	Best E	6.4946×10^{22}	2.1085×10^{14}	4.5255×10^{15}	8.8315×10^{13}	3.9199×10^{18}	2.7205×10^{22}
WOA	Avg. CF	5.09×10^{60}	7.1636×10^{69}	1.1383×10^{68}	6.548×10^{71}	4.6704×10^{71}	7.2743×10^{71}
	Best CF	4×10^{60}	1.4×10^{61}	1.8×10^{61}	3.6658×10^{71}	3.512×10^{71}	5.6986×10^{71}
	Avg. NP	5	15	20	31	37	44
	Best NP	4	14	18	26	32	38
	Avg. E	6.1491×10^{23}	5.6463×10^{14}	1.8316×10^{16}	1.2598×10^{14}	5.2791×10^{18}	3.392×10^{22}
	Best E	4.7187×10^{22}	1.895×10^{14}	8.9749×10^{15}	4.7129×10^{13}	2.0887×10^{18}	1.7534×10^{22}
ACO	Avg. CF	4.5699×10^{71}	3.7148×10^{71}	9.3099×10^{71}	1.4033×10^{72}	1.0814×10^{72}	1.4512×10^{72}
	Best CF	8×10^{60}	1.5×10^{61}	2.2×10^{61}	1.2746×10^{72}	1.0048×10^{72}	1.2674×10^{72}
	Avg. NP	4	7	8	9	10	12
	Best NP	2	4	5	6	7	6
	Avg. E	3.0953×10^{22}	4.0813×10^{13}	5.9113×10^{14}	1.751×10^{12}	3.6959×10^{16}	4.261×10^{20}
	Best E	4.4537×10^5	1.1508×10^{12}	1.3255×10^6	1.957×10^{10}	1.7838×10^{12}	7.7962×10^{18}
OOA	Avg. CF	4.96×10^{60}	4.7818×10^{70}	5.1587×10^{69}	7.7914×10^{71}	5.4469×10^{71}	8.1832×10^{71}
	Best CF	4×10^{60}	1.5×10^{61}	1.9×10^{61}	6.0594×10^{71}	4.679×10^{71}	6.7698×10^{71}
	Avg. NP	5	15	21	29	36	42
	Best NP	4	13	19	25	32	37
	Avg. E	9.5457×10^{23}	6.7351×10^{14}	2.2913×10^{16}	1.4039×10^{14}	5.9964×10^{18}	3.5683×10^{22}
	Best E	2.0688×10^{23}	2.1862×10^{14}	1.129×10^{16}	7.4459×10^{13}	2.9095×10^{18}	1.8181×10^{22}
MA	Avg. CF	4.43×10^{60}	1.516×10^{61}	2×10^{61}	6.5542×10^{71}	4.7063×10^{71}	7.2175×10^{71}
	Best CF	4×10^{60}	1.3×10^{61}	1.8×10^{61}	5.0182×10^{71}	3.8797×10^{71}	5.8847×10^{71}
	Avg. NP	4	15	20	31	38	43
	Best NP	4	13	18	28	34	37
	Avg. E	4.697×10^{23}	5.641×10^{14}	1.6752×10^{16}	1.2139×10^{14}	5.5114×10^{18}	3.3116×10^{22}
	Best E	9.1966×10^{22}	2.7977×10^{14}	7.8853×10^{15}	3.1976×10^{13}	3.5254×10^{18}	2.0607×10^{22}

Table 4. Cont.

Algorithm	Measure	Network 1	Network 2	Network 3	Network 4	Network 5	Network 6
AOA	Avg. CF	4.69×10^{60}	3.3716×10^{70}	1.0224×10^{70}	7.8676×10^{71}	5.4175×10^{71}	8.0266×10^{71}
	Best CF	4×10^{60}	1.4×10^{61}	1.8×10^{61}	5.593×10^{71}	4.2144×10^{71}	6.4571×10^{71}
	Avg. NP	5	15	20	28	34	40
	Best NP	4	12	17	23	29	34
	Avg. E	4.5233×10^{23}	2.8891×10^{14}	1.05×10^{16}	6.0699×10^{13}	2.7815×10^{18}	2.0416×10^{22}
	Best E	1.2327×10^{23}	7.5247×10^{13}	4.099×10^{15}	1.6334×10^{13}	8.9783×10^{17}	7.737×10^{21}
CHIO	Avg. CF	4.7×10^{60}	3.0251×10^{70}	1.4891×10^{70}	7.7324×10^{71}	5.5317×10^{71}	8.2538×10^{71}
	Best CF	4×10^{60}	1.4×10^{61}	1.8×10^{61}	4.2028×10^{71}	4.3911×10^{71}	6.7268×10^{71}
	Avg. NP	5	15	21	29	36	42
	Best NP	4	13	18	24	29	35
	Avg. E	4.7572×10^{23}	5.6382×10^{14}	1.576×10^{16}	1.0685×10^{14}	4.7282×10^{18}	3.0955×10^{22}
	Best E	1.0896×10^{23}	2.081×10^{14}	8.5226×10^{15}	4.7669×10^{13}	2.0199×10^{18}	1.8926×10^{22}
DTBO	Avg. CF	4.43×10^{60}	2.3611×10^{70}	4.6668×10^{69}	6.4274×10^{71}	4.5929×10^{71}	6.933×10^{71}
	Best CF	4×10^{60}	1.3×10^{61}	1.7×10^{61}	3.1498×10^{71}	2.9082×10^{71}	4.7492×10^{71}
	Avg. NP	4	16	19	30	38	45
	Best NP	4	13	17	25	31	37
	Avg. E	4.1237×10^{23}	5.4967×10^{14}	1.3802×10^{16}	1.1513×10^{14}	4.9397×10^{18}	3.2404×10^{22}
	Best E	1.5942×10^{23}	2.6794×10^{14}	7.1215×10^{15}	6.0937×10^{13}	3.0635×10^{18}	2.1081×10^{22}
PHAMD	Avg. CF	4×10^{60}	1.324×10^{61}	1.716×10^{61}	3.636×10^{61}	4.502×10^{61}	5.489×10^{61}
	Best CF	4×10^{60}	1.2×10^{61}	1.5×10^{61}	3.6×10^{61}	4.4×10^{61}	5.3×10^{61}
	Avg. NP	4	13	17	36	45	53
	Best NP	4	12	15	36	44	50
	Avg. E	3.6528×10^{23}	4.2558×10^{14}	1.2833×10^{16}	1.3892×10^{14}	6.647×10^{18}	4.0136×10^{22}
	Best E	3.4829×10^{22}	1.4022×10^{14}	6.1203×10^{15}	8.0683×10^{13}	4.0684×10^{18}	2.7623×10^{22}

The Average Cost Function (Avg. CF) represents the mean value of the cost function, indicating overall performance efficiency. The Best Cost Function Result (Best CF) shows the lowest cost function value achieved, reflecting the algorithm's capability to find optimal solutions. The Best Number of Pinning Nodes (Best NP) and the Average Number of Pinning Nodes (Avg. NP) provide insight into the algorithm's effectiveness in minimizing control nodes. Energy consumption is also evaluated, with the Best Energy Consumption (Best E) indicating the lowest energy usage, and the Average Energy Consumption (Avg. E) representing mean energy consumption.

To emphasize stability, values corresponding to unstable solutions are marked in red, indicating networks that did not achieve stability, while stable solutions are marked in black. This color-coding facilitates quick assessment of which algorithms balance energy efficiency with network stability. Overall, Table 4 underscores the PHA-MD algorithm's superior performance, particularly in achieving stable solutions with efficient energy use and minimal pinning nodes.

Diagnostic Visualization of the Node-Blocking Stage

To illustrate the internal mechanics of the proposed blocking stage—biologically, how *P. coffea* wasps cease to visit unpromising nodes—we include the following diagnostic plots (Figures 17–22). These figures are intended to explain how nodes are progressively blocked or excluded from the pinning set, rather than to provide additional performance benchmarks. For clarity, let n^* denote the current average number of successful visits across the remaining (non-blocked) nodes; nodes below this running threshold are provisionally excluded from the pinning set, while blocked nodes become members of $\mathcal{B}(t)$. If stability is still not achieved after blocking or excluding a node, that node can be returned to the candidate pool, consistent with the update rules in Section 3.2.

As shown in Figure 17, the first network begins without any blocked nodes, but several nodes lie below the threshold. Figure 18 presents the corresponding behavior for Network 2, while Figure 19 illustrates a topology that requires fewer pinning nodes. Figures 20–22 summarize the results for the remaining networks.

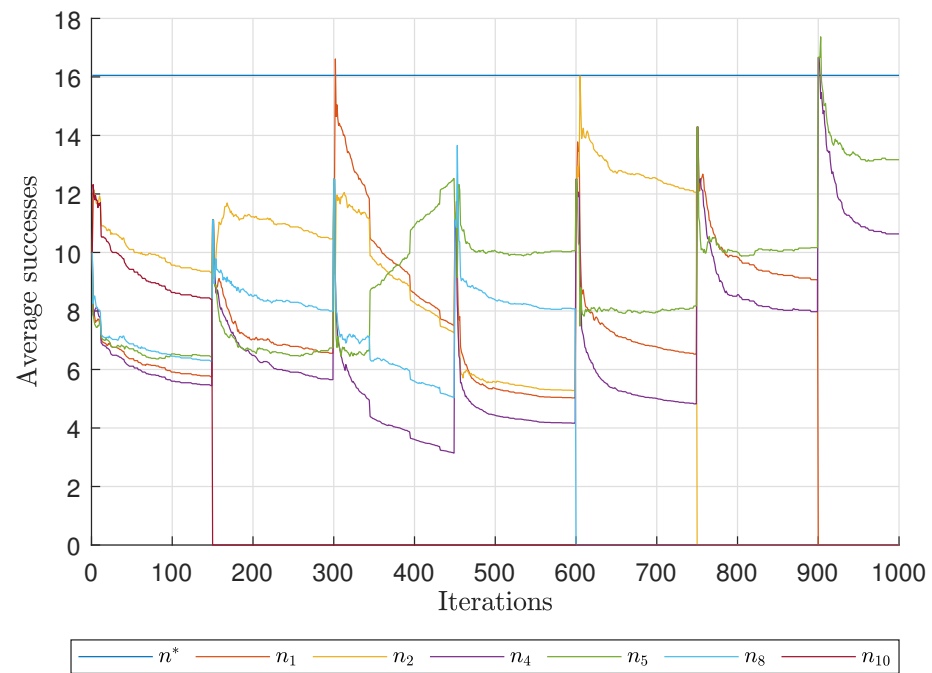


Figure 17. Blocked and excluded nodes for Network 1. Before the first epoch, there are no blocked nodes; however, several nodes lie below the performance threshold n^* . Between iterations 150 and 600, only one node is blocked, indicating that the population had not yet identified which node should be blocked to guarantee stability. After iteration 900, nodes n_{10} , n_8 , n_2 , and n_1 are blocked, and nodes n_4 and n_5 are excluded from the pinning set; the remaining nodes then ensure network stability.

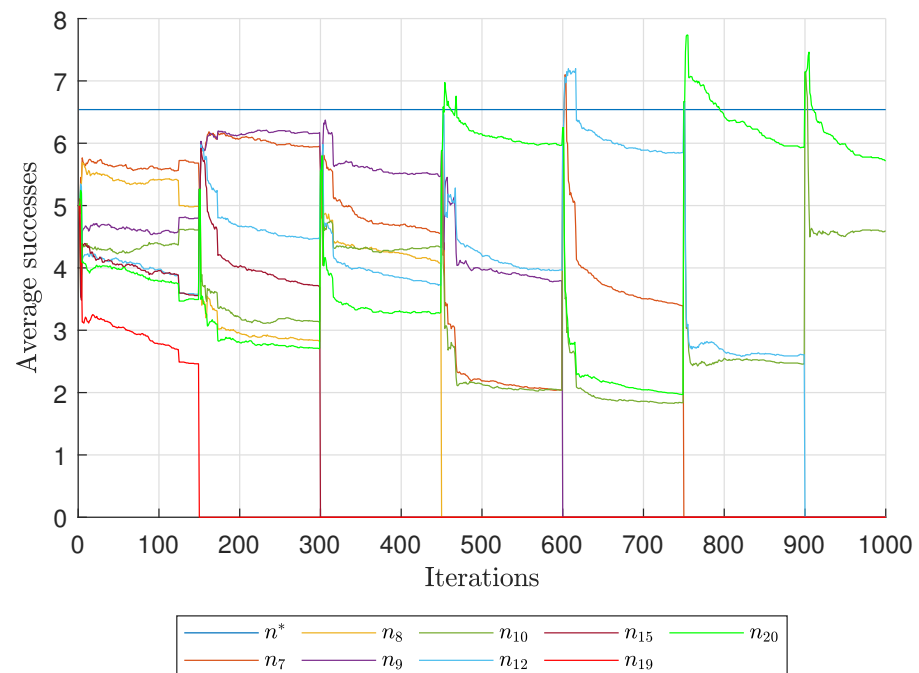


Figure 18. Blocked and excluded nodes for Network 2. Between iterations 150 and 900, the blocking set has a cardinality of six. The blocking node n_8 at iteration 750 causes the average number of successes of node n_{20} to exceed the threshold, yet node n_{12} is blocked at iteration 900 and nodes n_{20} and n_{10} are subsequently excluded from the pinning set.

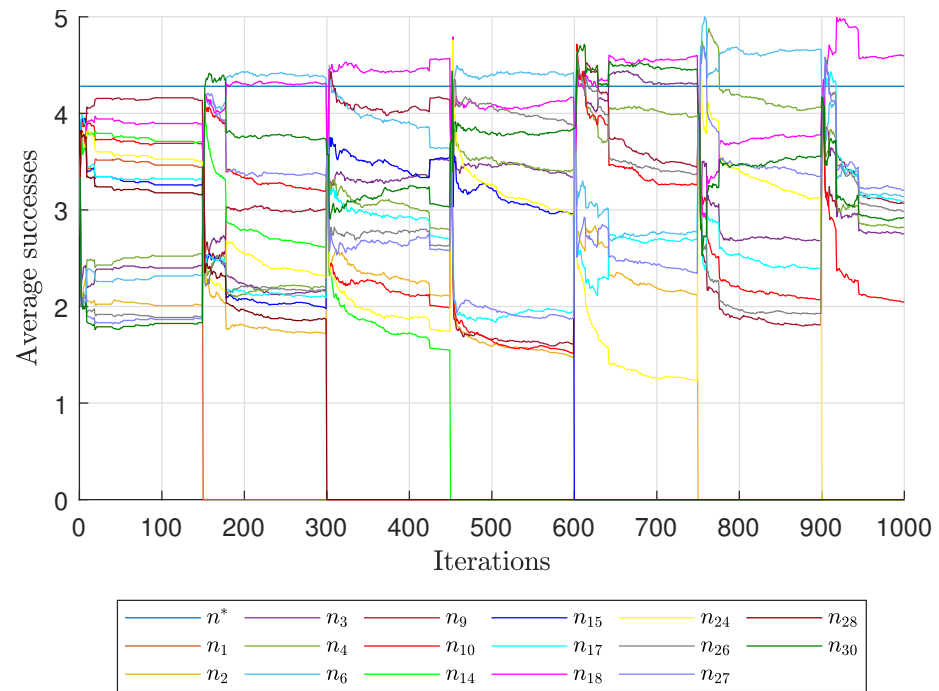


Figure 19. Blocked and excluded nodes for Network 3. Owing to the network topology and initial conditions, this case requires fewer pinning nodes than Network 2 for stabilization. Between iterations 900 and 1000, the excluded node n_{18} appears above the threshold n^* due to randomness and the local/global influence coefficients of the agents.

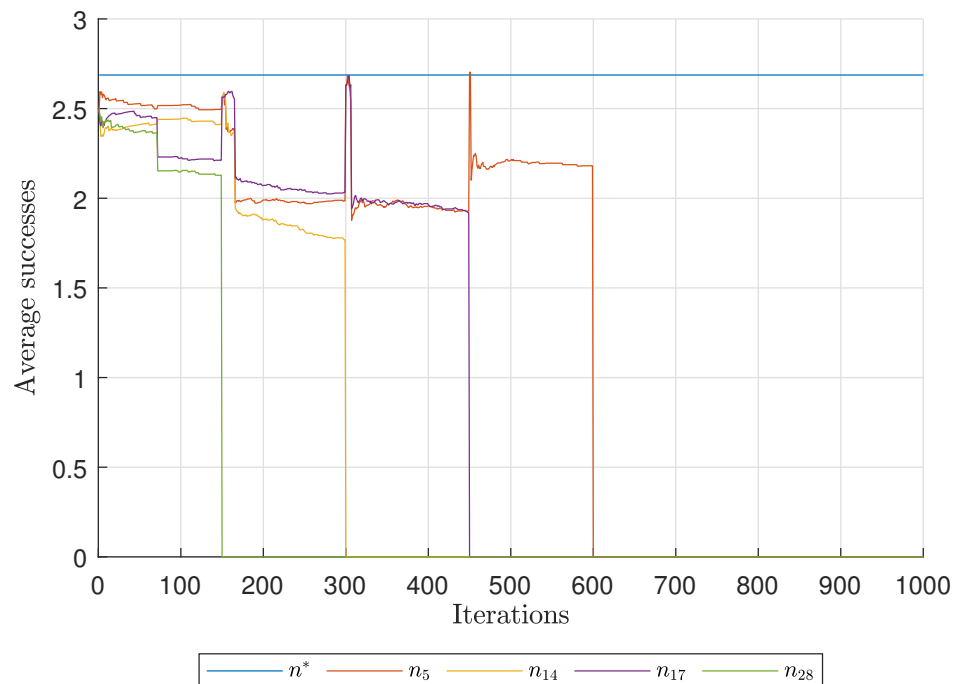


Figure 20. Blocked and excluded nodes for Network 4. In this case, the algorithm finishes the iterative process with blocked nodes; unlike previous simulations, Network 4 requires a high number of pinning nodes for stabilization, and the blocked nodes remain below the threshold n^* .

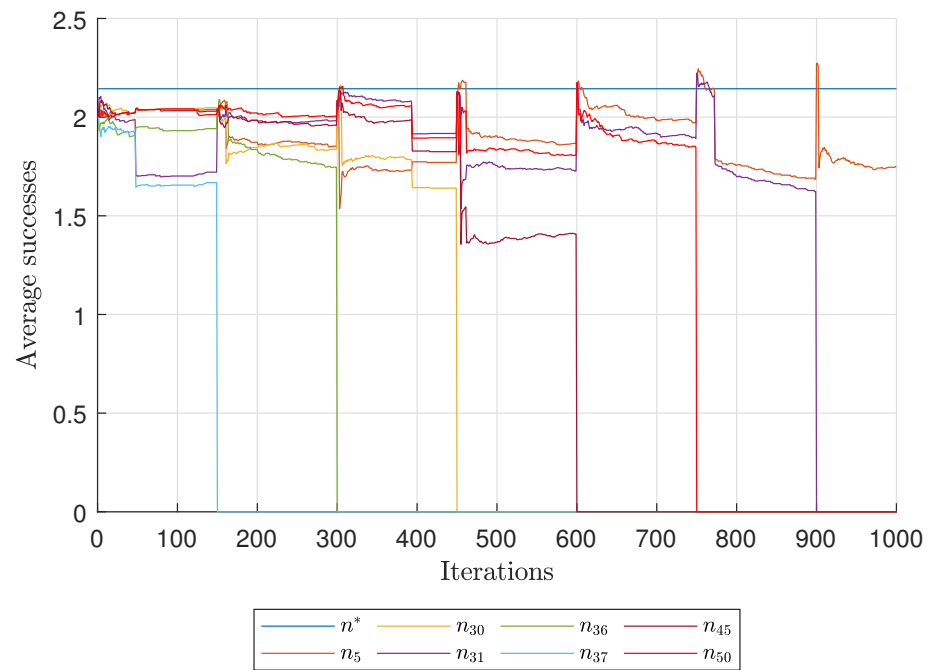


Figure 21. Blocked and excluded nodes for Network 5. This figure shows one of two permutations of the final blocked/excluded sets. PHA-MD may vary the order in which nodes are blocked or excluded; some nodes are excluded because they were previously blocked, and conversely, some are blocked because they were excluded. Therefore, solutions can differ depending on the blocking/exclusion order while still satisfying stability.

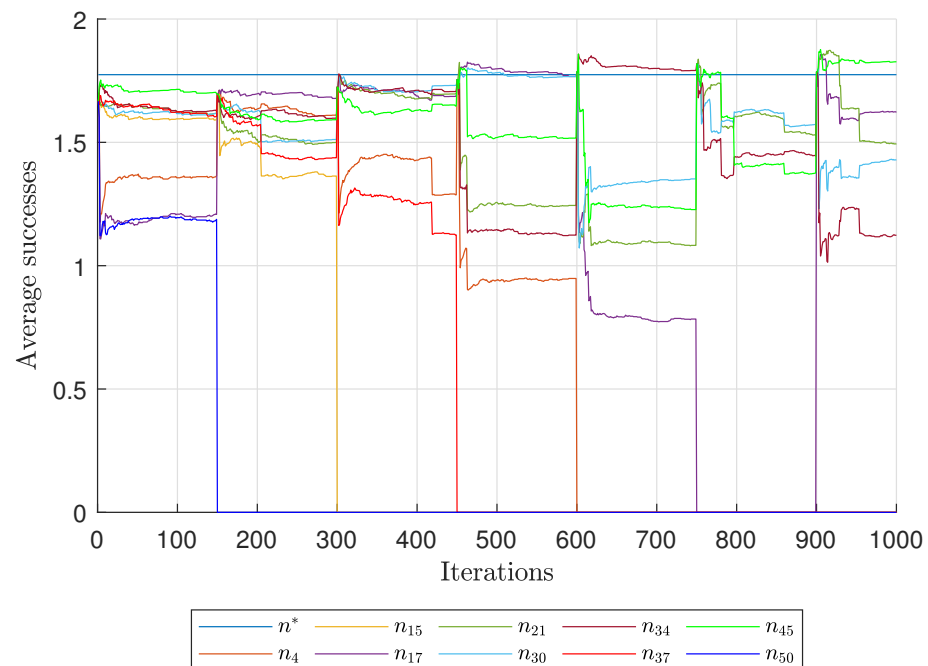


Figure 22. Blocked and excluded nodes for Network 6. If stability is not achieved after blocking or excluding a node, that node can return to the candidate set (e.g., node n_{17} at iteration 900).

5. Conclusions

The proposed PHA-MD algorithm is an effective method for pinning control of complex networks, eliminating the need for prior information on the minimal number of pinned nodes. It is designed to solve multi-constraint optimization problems, focusing on optimal node selection with the minimal number of pinned nodes, while ensuring network stabilization through the V-stability tool.

A stability-first evaluation has been adopted. In the pinning control setting, feasibility (stability) is the primary objective; therefore, candidates that are energy-efficient but unstable are not considered valid solutions. Cross-method comparisons have been reported under equal population and iteration budgets and with explicit instability flags, so as not to conflate algorithmic merit with implementation details or with external enumeration required by fixed-dimension heuristics. Within this framing, PHA-MD consistently identified smaller, stable pinning sets and competitive energy levels across increasingly challenging network topologies.

A key advantage of PHA-MD is its ability to handle variable dimensions for its agents, unlike current algorithms that operate with fixed dimensions. This adaptability allows PHA-MD to address challenges related to permutations in pinning node selection, making it a robust and versatile solution for complex network control. Consequently, the hybrid nature of the optimization problem tackled by PHA-MD complicates direct comparisons with other collaborative behavior algorithms.

Future research could explore various weighting methodologies in penalty functions, analyze convergence times in border functions, and investigate more complex decision vector dynamics. These studies could further enhance the algorithm's efficiency and applicability in more challenging scenarios, reinforcing its role as a leading solution in complex network control.

Author Contributions: Conceptualization, J.A.L. and A.Y.A.; methodology, J.A.L. and J.R.-L.; software, J.A.L. and A.J.P.; validation, R.C.-N. and C.L.-Á.; formal analysis, J.R.-L. and Y.A.-M.; investigation, J.A.L., A.J.P., and J.R.-L.; resources, C.L.-Á. and H.A.G.-O.; data curation, R.C.-N. and A.J.P.; writing—original draft preparation, J.A.L. and J.R.-L.; writing—review and editing, L.F.L.-V., A.Y.A., and Y.A.-M.; visualization, R.C.-N. and C.L.-Á.; supervision, Y.A.-M. and H.A.G.-O.; project administration, L.F.L.-V.; funding acquisition, C.L.-Á. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to the size and complexity of the raw simulation files.

Acknowledgments: The authors greatly appreciate Edgar Nelson Sánchez Camperos for allowing us to develop this project and helping us grow as research scientists.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PHA-MD	Phymastichus–Hypothenemus Algorithm—Minimized and Determinated
ALO	Ant Lion Optimizer
TLBO	Teaching–Learning-Based Optimization
GWO	Grey Wolf Optimizer
AMO	Animal Migration Optimization
PSO	Particle Swarm Optimization
ABC	Artificial Bee Colony
GSK	Gaining–Sharing Knowledge Based Algorithm
BBO	Biogeography-Based Optimization
WOA	Whale Optimization Algorithm

ACO	Ant Colony Optimization
OOA	Osprey Optimization Algorithm
MA	Mayfly Algorithm
AOA	Archimedes Optimization Algorithm
CHIO	Coronavirus Herd Immunity Optimizer
DTBO	Driving Training-Based Optimization
GA	Genetic Algorithm

References

- Barabási, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [\[CrossRef\]](#)
- Strogatz, S.H. Exploring complex networks. *Nature* **2001**, *410*, 268–276. [\[CrossRef\]](#) [\[PubMed\]](#)
- Newman, M.E.J. The structure and function of complex networks. *SIAM Rev.* **2003**, *45*, 167–256. [\[CrossRef\]](#)
- Alon, U. *An Introduction to Systems Biology: Design Principles of Biological Circuits*; Chapman and Hall and CRC: Boca Raton, FL, USA, 2006.
- Sporns, O.; Chialvo, D.R.; Kaiser, M.; Hilgetag, C.C. Organization, development and function of complex brain networks. *Trends Cogn. Sci.* **2004**, *8*, 418–425. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wasserman, S.; Faust, K. *Social Network Analysis: Methods and Applications*; Cambridge University Press: Cambridge, UK, 1994.
- Pimm, S.L. *Food Webs*; University of Chicago Press: Chicago, IL, USA, 1982.
- Motter, A.E.; Lai, Y.C. Cascade-based attacks on complex networks. *Phys. Rev. E* **2002**, *66*, 065102. [\[CrossRef\]](#)
- Buldyrev, S.V.; Parshani, R.; Paul, G.; Stanley, H.E.; Havlin, S. Catastrophic cascade of failures in interdependent networks. *Nature* **2010**, *464*, 1025–1028. [\[CrossRef\]](#)
- Freeman, L.C. Centrality in social networks conceptual clarification. *Soc. Netw.* **1978**, *1*, 215–239. [\[CrossRef\]](#)
- Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [\[CrossRef\]](#)
- Sabidussi, G. The centrality index of a graph. *Psychometrika* **1966**, *31*, 581–603. [\[CrossRef\]](#)
- Borgatti, S.P. Centrality and network flow. *Soc. Netw.* **2005**, *27*, 55–71. [\[CrossRef\]](#)
- Newman, M.E.J.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. [\[CrossRef\]](#)
- Zhang, H.; Liu, Y.; Wang, J. Adaptive control for complex networks: A review. *IEEE Trans. Control Netw. Syst.* **2020**, *7*, 1234–1245.
- Chen, G. Controllability robustness of complex networks. *J. Autom. Intell.* **2022**, *1*, 100004. [\[CrossRef\]](#)
- Hu, J.; Zhang, H.; Liu, H.; Yu, X. A survey on sliding mode control for networked control systems. *Int. J. Syst. Sci.* **2021**, *52*, 1129–1147. [\[CrossRef\]](#)
- Lizarraga, J.A.; Vega, C.J.; Sanchez, E.N. Particles swarm optimization for minimal energy consumption in complex networks node search. In Proceedings of the 2022 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 11–13 November 2020; pp. 1–6.
- Lizarraga, J.A.; Ruiz-Leon, J.; Sanchez, E.N. Phymastichus-hypothenemus-based algorithm for optimal node selection on pinning control of complex networks. In Proceedings of the 2022 19th International Conference on Electrical Engineering, Computing Science and Automatic Control, Mexico City, Mexico, 9–11 November 2022; pp. 1–6.
- Xiang, J.; Chen, G. On the V-stability of complex dynamical networks. *Automatica* **2007**, *43*, 1049–1057. [\[CrossRef\]](#)
- Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [\[CrossRef\]](#)
- Rao, R.; Savsani, V.; Vakharia, D. Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems. *Inf. Sci.* **2012**, *183*, 1–15. [\[CrossRef\]](#)
- Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [\[CrossRef\]](#)
- Li, X.; Zhang, J.; Yin, M. Animal migration optimization: An optimization algorithm inspired by animal migration behavior. *Neural Comput. Appl.* **2013**, *24*, 1867–1877. [\[CrossRef\]](#)
- Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN’95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
- Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [\[CrossRef\]](#)
- Wagdy, A.; Hadi, A.; Khater, A. Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1501–1529.
- Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [\[CrossRef\]](#)
- Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
- Dorigo, M.; Maniezzo, V.; Colnari, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 29–41. [\[CrossRef\]](#)

31. Sulaiman, M.; Saif, N.; Khader, A.T. Osprey Optimization Algorithm: A Novel Nature-Inspired Metaheuristic. *IEEE Access* **2022**, *10*, 54629–54641.
32. Zhang, T.; Zhou, Y.; Zhou, G.; Deng, W.; Luo, Q. Bioinspired Bare Bones Mayfly Algorithm for Large-Scale Spherical Minimum Spanning Tree. *Front. Bioeng. Biotechnol.* **2022**, *10*, 830037. [[CrossRef](#)]
33. Gandomi, A.H.; Yang, X.S.; Alavi, A.H. Archimedes Optimization Algorithm. *J. Comput. Des. Eng.* **2020**, *7*, 258–268.
34. Hatamlou, A. Coronavirus Herd Immunity Optimizer (CHIO): A novel metaheuristic algorithm for global optimization. *Comput. Ind. Eng.* **2021**, *161*, 107606.
35. Nasiri, M.; Mousavirad, S.M.; Mohammadi, S.R. Driving Training-Based Optimization (DTBO): A new optimization algorithm. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103570.
36. Holland, J. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
37. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
38. Chen, G.; Wang, X.; Li, X. *Fundamentals of Complex Networks: Models, Structures and Dynamics*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
39. Su, H.; Wang, X. *Pinning Control of Complex Networked Systems: Synchronization, Consensus and Flocking of Networked Systems via Pinning*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
40. Chen, G. Pinning control and synchronization on complex dynamical networks. *Int. J. Control Autom. Syst.* **2014**, *12*, 221–230. [[CrossRef](#)]
41. Bustillo, P.; Villagalba, G.; Chaves, C. Consideraciones sobre el uso de insecticidas químicos en la zona cafetera en el control de la broca del café, *hypothemus hampei*. *Congr. Soc. Colomb. Entomol.* **1993**, *20*, 152–158.
42. Espinoza, J.C. The biology of *phymastichus coffea lasalle* (hymenoptera: Eulophidae) under field conditions. *Biol. Control* **2009**, *49*, 227–233. [[CrossRef](#)]
43. Waller, J.M.; Bigger, M.; Hillocks, R.J. *Coffee Pests, Diseases and Their Management*; CAB Books; CABI Pub: Oxon, UK, 2007; Volume 1, pp. 70–71.
44. Cody, W.J. Algorithm 665: Machar: A subroutine to dynamically determined machine parameters. *ACM Trans. Math. Softw.* **1988**, *14*, 303–311. [[CrossRef](#)]
45. Leskovec, J.; Huttenlocher, D.; Kleinberg, J. Predicting Positive and Negative Links in Online Social Networks. In Proceedings of the 19th International Conference on World Wide Web (WWW '10), New York, NY, USA, 26–30 April 2010; pp. 641–650. [[CrossRef](#)]
46. Yizhar, O.; Fenno, L.E.; Davidson, T.J.; Mogri, M.A.; Deisseroth, K. Neocortical Excitation/Inhibition Balance in Information Processing and Social Dysfunction. *Nature* **2011**, *477*, 171–178. [[CrossRef](#)]
47. Karlebach, G.; Shamir, R. Modelling and analysis of gene regulatory networks. *Nat. Rev. Mol. Cell Biol.* **2008**, *9*, 770–780. [[CrossRef](#)]
48. Sun, J. Impedance-Based Stability Criterion for Grid-Connected Inverters. *IEEE Trans. Power Electron.* **2011**, *26*, 3075–3078. [[CrossRef](#)]
49. Kundur, P. *Power System Stability and Control*; McGraw-Hill: New York, NY, USA, 1994.
50. Allesina, S.; Tang, S. Stability criteria for complex ecosystems. *Nature* **2012**, *483*, 205–208. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.