Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Genetic algorithm-based secure cooperative control for high-order nonlinear multi-agent systems with unknown dynamics

Xin Wang[1], Dongsheng Yang[1*], D Raveena Judie Dolly[2], Shuang Chen[3], Madini O. Alassafi[4], Fawaz E. Alsaadi[4] and Jianhui Lyu[5]

Abstract

Research has recently grown on multi-agent systems (MAS) and their coordination and secure cooperative control, for example in the field of edge-cloud computing. MAS offers robustness and flexibility compared to centralized systems by distributing control across decentralized agents, allowing the system to adapt and scale without overhaul. The collective behavior emerging from agent interactions can solve complex tasks beyond individual capabilities. However, controlling high-order nonlinear MAS with unknown dynamics raises challenges. This paper proposes an enhanced genetic algorithm strategy to enhance secure cooperative control performance. An efficient encoding method, adaptive decoding schemes, and heuristic initialization are introduced. These innovations enable compelling exploration of the solution space and accelerate convergence. Individual enhancement via load balancing, communication avoidance, and iterative refinement intensifies local search. Simulations demonstrate superior performance over conventional algorithms for complex control problems with uncertainty. The proposed method promises robust, efficient, and consistent solutions by adapting to find optimal points and exploiting promising areas in the space. This has implications for securely controlling real-world MAS across domains like robotics, power systems, and autonomous vehicles.

Keywords Multi-agent systems, Genetic algorithm, Secure cooperative control, High-order nonlinear model, Unknown dynamics, Edge-cloud computing

Introduction

With the emerging of ChatGPT, the multiple model predictive control in the edge-cloud computing become considerably important. Correspondingly, the research landscape has recently been enriched by an increased focus on multi-agent systems (MAS). This burgeoning interest is especially pronounced in agents' coordination and secure cooperative control. At its core, the study of MAS revolves around a simple yet profound idea: equipping follower agents with the capability to synchronize their actions or match the state trajectories of a leader agent based solely on local interactions [1]. Why is this shift towards MAS when conventional systems reliant on single centralized controllers have been prevalent? The answer lies in the advantages distributed MAS offers over

*Correspondence:
Dongsheng Yang
yangdongsheng@mail.neu.edu.cn
[1] College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[2] Department of Electronics and Communication Engineering, Karunya Institute of Technology and Sciences, Tamil Nadu, India
[3] Dongneng (Shenyang) Energy Engineering Technology Co.,Ltd, Shenyang 110069, China
[4] Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[5] Tsinghua University, Beijing 100084, China

their centralized counterparts [2]. Traditional centralized control systems operate from a singular point of command, potentially making them vulnerable to failures or system-wide disruptions. In contrast, MAS are inherently decentralized and thus provide a heightened sense of robustness. The decentralized nature ensures the system remains operative even if some agents malfunction or get compromised [3].

Beyond robustness, MAS exhibits flexibility, an attribute that's becoming ever more crucial in today's fast-evolving technological landscape. This flexibility stems from the system's adaptability, allowing for changes or additions without a comprehensive overhaul [4]. For instance, as new agents are introduced into the system, they can be seamlessly integrated, leveraging the existing network of interactions. Scalability, another cardinal advantage of MAS, ensures that these systems can grow in size and complexity without a proportionate increase in management overhead, which is particularly beneficial in scenarios that require large-scale operations or foresee growth in the number of agents or tasks [5, 6]. By virtue of their design, Distributed MAS can efficiently distribute tasks among agents, making them particularly adept at parallelizing complex problems. One of the most intriguing facets of MAS is the emergence of sophisticated group-level behaviors [7]. While each agent in the system might be programmed for only a specific set of simple tasks, the collective behavior that emerges when they interact can be unexpectedly intricate. This phenomenon of self-organization is reminiscent of natural systems, where simple entities come together to produce behaviors far beyond their capabilities. Ant colonies, for example, demonstrate complex cooperative behaviors despite having limited actions.

Over the years, the predominant focus in MAS has been on idealized first-order and second-order linear models [8]. However, most real-world agents' true complexity and richness lie in nonlinear dynamics characterized by higher dimensionality [9]. This observed disparity between academic models and practical realities prompts a deeper exploration into more intricate systems and their associated behaviors. For example, robotic systems. Unlike the rudimentary linear models, these systems are defined by nonlinear coupled joint kinematics and dynamics. Their movements and interactions are a culmination of complex algorithms that govern each joint, each sensor, and each actuator. Similarly, biological agents such as animals present an intricate dance of neural and musculoskeletal dynamics. Rather than linearly responding to stimuli, these agents engage in a sophisticated, multi-tiered interplay of cognitive, neural, and physical processes. The complexities do not end here. Power grids are pivotal to our modern infrastructure and do not operate on linear principles. Instead, they are characterized by nonlinear oscillatory dynamics, with each component responding to fluctuations in a manner that often diverges from basic linear expectations. Meanwhile, multi-vehicle systems, aerial drones, or undersea exploratory vehicles demonstrate intricate rigid body motions that do not conform neatly to linearized models. Venturing into socio-economic dimensions, consumer markets and social networks offer myriad challenges. These systems are governed not by mechanical or biological rules but by the unpredictable, often contradictory, whims and behaviors of human agents. Their patterns, although sometimes discernible, are rooted in the heterogeneous nature of human behavior, making them far removed from the simplicity of first or second-order linear models.

Given these complexities, the realm of MAS control has seen a myriad of methodologies being introduced and refined [10]. Techniques such as consensus algorithms serve to streamline decision-making processes across agents. Optimization-based techniques prioritize finding the best solutions within certain constraints. Simultaneously, biologically inspired approaches seek to harness the elegant solutions nature has honed over millennia. However, several issues need to be addressed when applying genetic algorithm (GA) to cooperative control in MAS. First, existing studies in this area often focus on simple agent dynamics, such as single or double integrators. The cooperative control of high-order nonlinear MAS needs to be investigated, leaving a gap in understanding how GA can be effectively applied in such scenarios. Second, the encoding method used in GA can limit the completeness of the solution space and the efficiency of the search process. Typical representations for ordering problems may need help to handle the complexity of controlling topology in MAS [11]. This limitation can hinder the ability of the GA to explore the entire solution space and find optimal solutions. Additionally, premature convergence and weak local search capability of GA can lead to suboptimal performance in cooperative control problems.

In conclusion, the study of MAS and their coordination and cooperative control has received significant attention in recent years. GA has shown promise in this field due to their global search capability. However, limitations exist in applying GA to cooperative control, such as considering simple agent dynamics, limitations in encoding methods, premature convergence, and weak local search capability. Specifically, it now discusses how typical GA encodings restrict the completeness of the solution space, slowing convergence for ordering problems. The lack of adaptive decoding schemes in GA is also a weakness when handling complex MAS control topologies. Finally, the issues of premature convergence

and weak local search hamper existing GA performance. These gaps motivate the improved GA proposed to overcome said limitations. Therefore, this paper proposes an improved GA-based strategy that addresses these limitations by considering high-order nonlinear dynamics, introducing an efficient encoding method, developing adaptive decoding and individual enhancement schemes, and employing a heuristic initialization technique. The proposed strategy demonstrates superior performance through simulations, showcasing its potential for cooperative control in MAS.

The proposed approach offers several distinct contributions compared to previous works.

- An improved GA is proposed for secure cooperative control of high-order nonlinear MAS with unknown dynamics. Simultaneously, an efficient encoding method is introduced to represent both topology control and execution order, the adaptive decoding scheme is developed using heuristics to accelerate convergence, and individual enhancement techniques like load balancing, communication avoidance, and iterative refinement are applied to intensify local search.
- A resilient control strategy is developed against false data injection attacks via attack estimation and compensation.

The rest of the paper is organized as follows. In Related works section, the related works are studied. In High-order nonlinear MAS section, the high-order nonlinear MAS is presented in detail. In Methodology section, the methodology is extended. Simulation section presents the simulation results, and the conclusions are drawn in Conclusions section.

Related works

MAS and cooperative control

MAS has emerged as a dominant paradigm in control systems and robotics, reflecting the collective efforts of multiple autonomous agents working towards a shared or individual objective [12, 13]. These agents, defined by their abilities to perceive, act, and make decisions, have been integral to several applications ranging from autonomous vehicle fleets to distributed sensing networks. Their significance lies in their number and how they interact, share information, and cooperate. Cooperative control, as a concept, complements MAS by focusing on the coordinated efforts of these agents [14–16]. This coordination ensures that the overall system achieves a particular behavior or objective more efficiently than individual agents working in isolation. Historically, cooperative control has been pivotal in solving complex problems that are otherwise beyond the reach of single-agent systems. The synergy between MAS and cooperative control has resulted in a transformative shift, particularly in swarm robotics, decentralized power grids, and intelligent transportation systems. In [17], a new technique based on complex Laplacian was introduced to address the problems of which formation shapes specified by inter-agent relative positions could be formed and how they could be achieved with distributed control ensuring global stability. Literature [17] introduced complex Laplacian method for distributed formation control and provided theoretical stability guarantees but limited validation. In [18], the authors investigated the robust cooperative output regulation problem of uncertain linear MAS with additive disturbances via the celebrated internal model principle. Literature [18] proposed robust distributed control for consensus tracking and handled uncertainties but high communication overhead. While the potential of MAS in conjunction with cooperative control is vast, it brings forth unique challenges [19]. These range from ensuring real-time communication between agents, making collective decisions in uncertain information, and developing adaptive strategies to handle dynamic environments and unexpected disruptions.

GA in control systems

GA has become a critical player in optimization and search problems inspired by natural selection. With roots in evolutionary biology, GA approaches problems by simulating processes like mutation, crossover, and selection, which are fundamental to biological evolution. Within control systems, the marriage of GA with traditional methods has given rise to innovative solutions, especially in scenarios where the search space is vast or poorly understood [20]. Literature [20] optimized active suspension with GA-PID and offered good convergence but fixed encoding limits exploration. The adaptability and versatility of GA have proven invaluable in automatically tuning controllers, optimizing system parameters, and even in the design of control strategies from the ground up. GA often provides robust and efficient solutions when conventional mathematical models falter due to complexities or non-linearity. In [21], the authors introduced a combined method called PSO-GA to tackle constrained optimization issues, using particle swarm optimization (PSO) to enhance the vector while the GA modifies decision vectors through genetic processes. The equilibrium between exploration and exploitation is further refined by integrating crossover and mutation genetic operators into the PSO algorithm. Constraints in the problem are managed using a parameter-free penalty function. PSO-GA handled constraints well, while lacking mechanisms to prevent premature convergence. In [22], the authors presented a new feature selection method called ABACO, grounded in ant colony optimization (ACO). In

this approach, features are represented as interconnected graph nodes. Each primary node in this graph is divided into two sub-nodes: one to select and another to deselect the feature. The ACO is employed to choose nodes, with the condition that ants must traverse all features. ABACO selected features effectively, while encoding not tailored for ordering problems. A notable strength of GA in control systems is their inherent ability to handle multiple objectives simultaneously, making them well-suited for systems where trade-offs between different performance metrics are inevitable [23–25]. This quality is particularly relevant in modern control challenges, where objectives like energy efficiency, robustness, and rapid response might be in contention.

Security challenges in cooperative control

In the realm of MAS, cooperative control has emerged as a pivotal paradigm, enabling agents to achieve specific objectives collaboratively. However, the inherently interconnected nature of these systems introduces a spectrum of security challenges that threaten the integrity, availability, and confidentiality of communications and operations [26, 27]. Literature [26] addressed DoS attacks, while performance guarantees require ideal models. Literature [27] handled deception attacks but high communication overhead for synchronization. One intrinsic challenge stems from the distributed nature of MAS. With agents operating in tandem, relying on local interactions and shared information, the system is particularly vulnerable to misinformation. An adversarial or compromised agent within the system can disseminate misleading information, causing potential systemic failures or unintended behaviors. Moreover, the communication channels bridging these agents become focal points of vulnerability. Eavesdropping, for instance, can jeopardize the confidentiality of exchanged data, revealing strategic insights to potential adversaries. Man-in-the-middle attacks, where an attacker intercepts and possibly alters the communication between two agents, can disrupt operational continuity and manipulate agents into undertaking hazardous actions. Additionally, as agents often rely on consensus algorithms to achieve a unified decision or action, attacks targeting these algorithms can stall or disrupt the consensus process. By presenting false data or through a Sybil attack (where one adversary controls multiple nodes), an attacker can influence the consensus decision in their favor. Ensuring data security and privacy is pivotal in both domains to safeguard the interests and data of the entities involved, whether they are system agents or recommendation service users. In [28], the authors proposed an interaction-enhanced and time-aware graph convolution network for successive point-of-interest (POI) recommendation. The proposed method incorporates high-order relationships between users and POIs, improving recommendation accuracy. However, implementing the method and managing the associated data can be complex and resource-intensive. In [29], the authors offered a point-of-interest category recommendation model based on group preferences. The proposed model leverages group preferences, considering the influence of similar users to enhance recommendation accuracy. Focusing on POI categories instead of individual POIs makes efficient use of data to mine user interests. Nevertheless, the proposed model's performance heavily relies on the availability and quality of user check-in data and the accuracy of group categorization. Additionally, implementing locality-sensitive hashing and long short-term memory can be computationally intensive and pose scalability challenges for large datasets or real-time applications. In [30], the authors proposed a novel privacy-preserving point-of-interest recommendation model that incorporated users' privacy preferences based on a simplified graph convolutional neural network. The proposed model addresses privacy leakage in location-based social networks while providing POI recommendations. The simplified graph convolutional network analyzes high-order connectivity between users and POIs, enhancing recommendation accuracy. However, implementing generative models and graph convolutional networks can be complex and computationally intensive. While the model aims to protect user privacy, there may still be potential privacy trade-offs and vulnerabilities not accounted for in the evaluation. Most importantly, the model may be highly specialized for geological tourism and not easily generalize to other domains. Furthermore, the scalability of MAS makes them a prime target for distributed denial of service attacks [31]. By flooding the system with superfluous requests or messages, attackers can overload agents and communication channels, rendering the system non-responsive or slowing down its operational pace. Lastly, as the landscape of technology and cyber threats is continually evolving, the unpredictability of new, sophisticated attack vectors remains a looming challenge. MAS, especially those operating in dynamic and unstructured environments, must be prepared for unforeseen threats, necessitating a proactive and adaptive security framework.

High-order nonlinear MAS

MAS, consisting of interconnected autonomous agents, has emerged as an active research area with broad applications, including robotic teams, sensor networks, transportation systems, power grids, and biological systems. The key advantages of MAS are robustness, scalability, and distributed capabilities. Complex, large-scale tasks can be accomplished efficiently by coordinating

multiple simple agents. The agents leverage local information exchange with neighbors to achieve global team objectives.

An agent is an autonomous entity with computation, communication, and control capabilities. The agent state evolves based on its dynamics. Agents interact with neighbors to share information and coordinate actions. A graph captures the communication topology. The objective is to achieve team-level goals like consensus, formation, coverage, and flocking by designing distributed control protocols for each agent. A differential equation describes the dynamics of an individual agent:

$$\dot{x}_i(t) = f_i(x_i, u_i) \tag{1}$$

where $x_i(t) \in \mathbb{R}^n$ is the state vector, $u_i(t) \in \mathbb{R}^m$ is the control input, and $f_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the dynamics. Equation (1) shows the dynamics of agent i, and is a general nonlinear equation capturing complex motions. The general form encapsulates a wide range of agent models. For instance, for a vehicle, x_i could denote position and orientation, u_i is the steering/throttle commands, and f_i captures the rigid body kinematics relating acceleration to force and torque inputs. For a robot, x_i would contain the joint angles and velocities, u_i specifies the motor torques, and f_i represents the coupled joint dynamics. The motivation is to allow vast agent types with minimal assumptions on structure.

For an N agent system, the overall dynamics are:

$$\dot{X}(t) = F(X, U) \tag{2}$$

where $X = [x_1^T, \ldots, x_N^T]^T$ concatenates all states, $U = [u_1^T, \ldots, u_N^T]^T$ are the inputs, and F combines the individual dynamics f_i. Equation (2) aggregates the individual dynamics into a combined system model. X and U concatenate the states and inputs across all N agents. F combines the individual f_i models into an overall mapping and provides a holistic model by merging the agents' dynamics. The motivation is to analyze the collective multi-agent behavior for properties like synchronization and consensus.

The interaction topology is modeled as a graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$ where $\mathcal{V} = 1, \ldots, N$ is the node set representing agents, and $\varepsilon \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set of communication links.

There are two common objectives: consensus and synchronization. Consensus means all agents converge to a common state value:

$$\lim_{t \to \infty} \left| x_i(t) - x_j(t) \right| = 0, \forall i, j \tag{3}$$

Equation (3) defines the consensus objective mathematically and states that the difference between any two agents' states should converge to zero asymptotically, aligning all agents to a typical value. The motivation is to derive a quantifiable metric for evaluating control strategies. It imposes the key requirement that agents reach an agreement through local interactions.

While synchronization tracks a dynamic leader $x_0(t)$:

$$\lim_{t \to \infty} |x_i(t) - x_0(t)| = 0, \forall i \tag{4}$$

Equation (4) defines the leader-follower synchronization objective, where agents track a dynamic leader $x_0(t)$. It states that the deviation between agents and leaders should vanish, aligning agents with their evolving trajectory. The motivation is to provide a metric for assessing synchronization performance in leader-following paradigms.

Consensus means all agents converge to a common state value, mathematically defined as the difference between any two agents' states approaching zero as time goes to infinity. Synchronization, conversely, means all agents track a dynamic leader, mathematically defined as the difference between any agent's state and the leader's state approaching zero as time progresses.

Many initial MAS studies focused on simple first-order linear models:

$$\dot{x}_i = u_i \tag{5}$$

Equation (5) shows a simplified single integrator model where the control u_i directly commands velocity/acceleration. The linear case permits graph theory tools to be applied but lacks realism. The motivation is to establish a baseline before considering more complex dynamics.

The control input is directly the acceleration or velocity command. This simplified case applies graph theory and linear system analysis tools. The protocol can achieve consensus:

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij} \left(x_j - x_i \right) \tag{6}$$

where $\mathcal{N}_i$ are the neighbors of agent i and a_{ij} are graph edge weights. Stability can be proven using properties of the Laplacian matrix. First-order linear models must improve their ability to capture real-world agent dynamics. They apply graph theory and linear system analysis tools, and the control input is directly the acceleration or velocity command. While these models provide insights, they only partially represent the complexities of real-world agent dynamics. Higher-order and nonlinear MAS offer a more realistic representation, such as the second-order nonlinear system, capturing the intricacies of real-world dynamics more effectively.

Equation (6) gives a consensus protocol that drives all agents to the group average by leveraging neighbor-state differences. The weights a_{ij} encode the interaction graph topology. Stability can be proven using

properties of the graph Laplacian matrix. The motivation is to derive a distributed control rule that achieves consensus for first-order models, providing a foundation for more advanced extensions.

A more realistic model is a second-order nonlinear system:

$$\ddot{x}_i = f_i(\dot{x}_i, x_i, u_i) \tag{7}$$

where $x_i \in \mathbb{R}^n$ could represent position, $\dot{x}_i \in \mathbb{R}^n$ is velocity, and $u_i \in \mathbb{R}^n$ is an acceleration or force command.

Equation (7) presents a more realistic second-order nonlinear dynamics capturing intricacies like coupled physics and actuator constraints, which better represent real MAS like robot teams or autonomous vehicles. The motivation is to move beyond idealistic linear models to handle complex dynamics with techniques like Lyapunov methods or neural networks.

Many physical systems are accurately described by second-order nonlinear dynamics. Examples include:

- Vehicles: $\ddot{p}_i = f_i(v_i) + g_i(p_i)u_i$
- Robots: $\ddot{q}_i = h_i(\dot{q}_i, q_i) + M_i(q_i)^{-1}u_i$, showing the dynamics of a robotic manipulator, where q_i is the joint angle, M_i is the inertia matrix, and u_i is the control input. The nonlinear couplings between joints are captured through h_i.
- Oscillators: $\ddot{\theta}_i = f_i(\dot{\theta}_i, \theta_i) + u_i$, modelling an oscillator agent where θ_i is the phase and f_i characterizes the nonlinear oscillations. u_i provides control authority.

where p_i, v_i are position and velocity, q_i, M_i are joint angles and inertia matrix, and θ_i is phase.

Consensus protocols can be extended using acceleration feedback:

$$u_i = -\sum_{j \in \mathcal{N}_i} a_{ij}\left(\ddot{x}_i - \ddot{x}_j\right) \tag{8}$$

However, the stability analysis becomes more complex for nonlinear dynamics. Lyapunov and LaSalle's theorems are commonly used. While more realistic, second-order models are still low-dimensional compared to real MAS, which motivates high-order nonlinear models.

In many MAS applications, the agents have complex nonlinear dynamics with higher dimensionality:

$$\dot{x}_i = f_i(x_i, u_i), x_i \in \mathbb{R}^n, n > 2 \tag{9}$$

Some examples include:

- Aerial vehicles: 12 + states for rigid body dynamics.
- Robots: n joint angles/rates for an n-degrees of freedom manipulator
- Power systems: generator rotor angles and frequencies.
- Biological networks: large-scale neuronal population models.

The distributed control problem becomes very challenging for high-order nonlinear MAS.

Consensus depends on coordinated tuning of multiple states and inputs. Stability analysis is complicated with complex interconnected dynamics. Many applications have partially unknown models with uncertainties, disturbances, and unmodeled dynamics:

$$\dot{x}_i = f_i(x_i, u_i) + d_i(t) \tag{10}$$

where $d_i(t)$ represents uncertainties. This further complicates controller design and convergence guarantees.

Beyond consensus, high-order nonlinear MAS give rise to other multi-objective coordination problems such as (i) Formation control: achieving desired geometric patterns; (ii) Coverage: distributing agents over a region; (iii) Flocking: aligning agent velocities like bird swarms; (iv) Synchronization: track dynamic leaders.

While high-order nonlinear dynamics are considered, even these are simplified representations compared to the full complexity exhibited in real-world MAS. For instance, the dynamics of autonomous robotic systems can encompass intricate sensorimotor control loops, planning under uncertainty, and complex physical interactions. Power grid systems display complex oscillatory dynamics as well as stochastic perturbations. The threats considered, including false data injection attacks, also represent a subset of what systems may face in practice. The modeling can incorporate higher dimensionality, heterogeneous dynamics, and detailed physics-based interactions to extend the approach to more complex dynamics. The GA's adaptive decoding schemes can help handle the increased dimensionality and maintain efficacy despite uncertainties. For sophisticated threat models, the resilience strategies must expand beyond attack detection and mitigation to include adaptive security mechanisms and online learning of new threats.

Methodology

Consider a leader-following MAS consisting of N follower agents labeled 1 to N, and one leader agent labeled 0. The dynamics of follower i is modeled by the nonlinear continuous-time system:

$$\dot{x}_i(t) = f_i(x_i) + g_i(x_i)u_i(t) + d_i(t)y_i(t) = h_i(x_i) \tag{11}$$

where $x_i \in \mathbb{R}^n$ is the state vector, $u_i \in \mathbb{R}^m$ is the control input, $y_i \in \mathbb{R}^p$ is the measurable output, $f_i : \mathbb{R}^n \to \mathbb{R}^n$ and

$g_i : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are unknown smooth nonlinear functions representing the agent dynamics, $d_i(t) \in \mathbb{R}^n$ is the external disturbance input, and $h_i : \mathbb{R}^n \to \mathbb{R}^p$ is the output function.

The leader agent's dynamics are similarly given by:

$$\dot{x}_0(t) = f_0(x_0, t) y_0(t) = h_0(x_0) \tag{12}$$

where the leader's dynamics $f_0 : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ are also uncertain.

The control objective is to design distributed control protocols for each follower agent using local neighbor information, such that the followers cooperatively track the dynamic leader. In other words, the tracking error between each follower output y_i and the leader output y_0 should converge to zero:

$$\lim_{t\to\infty} ||y_i(t) - y_0(t)|| = 0, \forall i \in 1, \ldots, N \tag{13}$$

This cooperative tracking should be achieved despite the complex unknown nonlinear dynamics of both the leader and followers.

The interaction topology between the agents is modeled as a fixed directed graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$. The vertex set $\mathcal{V} = 0,1,\cdots,N$ contains all agents, with 0 denoting the leader. The edge set $\varepsilon \subseteq \mathcal{V} \times \mathcal{V}$ represents the communication links between agents. An ordered edge $(j, i) \in \varepsilon$ means agent i can receive information from agent j (but not necessarily vice versa).

The adjacency matrix $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{(N+1)\times(N+1)}$ associated with $\mathcal{G}$ is defined as:

$$a_{ij} = \begin{cases} w_{ij} > 0 \text{ if } (j,i) \in \varepsilon \\ 0 \quad \text{otherwise} \end{cases} \tag{14}$$

where w_{ij} denotes the weight of edge (j, i). The neighbor set $\mathcal{N}_i$ of each agent i is given by:

$$\mathcal{N}_i = j \in V | (j,i) \in \varepsilon \tag{15}$$

The graph Laplacian matrix $L = \left[l_{ij}\right] \in \mathbb{R}^{(N+1)\times(N+1)}$ captures the interaction topology:

$$l_{ij} = \begin{cases} \sum_{k=0}^{N} a_{ik} & \text{if } i = j \\ -a_{ij} & \text{if } i \neq j \end{cases} \tag{16}$$

This mathematical graph representation will be leveraged to analyze the stability and convergence properties.

A key aspect of applying GA is designing an appropriate encoding to represent control solutions (chromosomes). Here, each chromosome is encoded as a string of N random keys:

$$\text{chrom} = [k_1, k_2, \ldots, k_N] \tag{17}$$

where each key $k_i \in \mathbb{R}$.

The integer part $\lfloor k_i \rfloor$ encodes the controlling topology, specifically the agent that follower i connects its controlling input to. This captures the interaction graph edges.

The decimal part $k_i - \lfloor k_i \rfloor \in [0,1)$ encodes the execution priority of agent i, which determines the order in which agents are scheduled. A smaller key value indicates higher priority.

Compared to standard binary encoding, this proposed scheme has several advantages: (i) It provides a natural unified representation for topology control and execution ordering, which are critical to the cooperative tracking problem. (ii) Keys can be easily ordered to determine scheduling priorities. Binary strings lack a direct ordering relationship. (iii) The compact representation as real numbers is more efficient than binary bits. (iv) No unique encoding/decoding is needed to maintain precedence feasibility, unlike ordering problems. (v) The randomized keys thoroughly explore the complete solution space. Therefore, the proposed encoding successfully incorporates the two essential aspects of topology and priority in a simple yet effective way.

To generate a control solution, each chromosome must be decoded to obtain the actual controller parameters and execution schedule. The proposed decoding first sorts all the agents in ascending order of their keys:

$$k_{i_1} < k_{i_2} < \cdots < k_{i_N} \tag{18}$$

This priority ordering determines the relative execution schedule.

Next, each agent is allocated to the specific controller given by the integer part of its key:

$$con(i_k) = \lfloor k_{i_k} \rfloor, k = 1, \ldots, N \tag{19}$$

where $con(i)$ returns the controller assigned to agent i.

The complete decoding procedure is summarized in Algorithm 1 below:

```
Begin
  sort agents as [i_1, i_2, ..., i_N] by ascending key values
  for k = 1 to N do
    allocate agent i_k to controller [k_{i_k}]
  end-for
  return control topology and priority schedule
End
```

Algorithm 1. Decoding algorithm

This transforms a given chromosome into a valid control solution. The complexity is $O(N\log N)$ due to the sorting operation.

An example is shown in Fig. 1 to illustrate the decoding process. There are $N = 5$ follower agents. The chromosome encodes a random key for each agent. First, the agents are reordered by ascending keys as A3, A2, A5, A1, A4. Then the controllers are allocated from the integer parts - A3→C0, A2→C1, A5→C1, A1→C2, A4→C3.

The fitness function evaluates the quality of each control solution. It maps a chromosome to a numeric cost value. Here the fitness is defined based on the worst case tracking error:

$$J = \max_i ||e_i(t)||, \forall t \in [0, T] \tag{20}$$

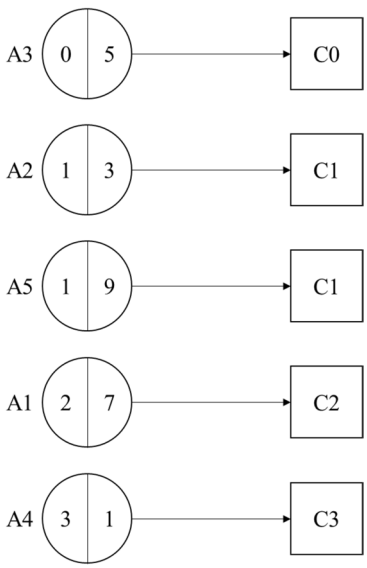

Fig. 1 Example decoding process from chromosome to control solution

where $e_i(t) = y_i(t) - y_0(t)$ is the tracking error for follower i between its own output and the leader's output.

A smaller fitness value indicates better tracking performance, driving the GA toward solutions that minimize the maximum deviation from the leader.

Computing the fitness requires simulating the agent dynamics under the control policy defined by the chromosome decoding. The fitness calculation has complexity $O(N)$.

The proposed approach introduces two additional heuristic decoding methods and the standard decoding procedure above to accelerate convergence.

- Earliest finish time (EFT) decoding: schedule agents according to the priority order based on key values and assign each agent to the controller, minimizing its finish execution time [32].
- Critical path (CP) decoding: first, rearrange agents in decreasing order of upward rank and then allocate controllers according to integer parts of keys [33].

The upward rank r_i of agent i indicates the length of the longest path from i to the leader, accounting for execution times and priorities:

$$r_i = \begin{cases} 0 & \text{if } i \text{ is the leader} \\ \max\limits_{p \in P_i} r_p + w_{pi} & \text{otherwise} \end{cases} \tag{21}$$

where P_i is the set of predecessors of i in the graph.

The motivation is that heuristic methods can more quickly converge to high-quality areas of the solution space. EFT minimizes individual finish times, while CP optimizes the critical upward path. To balance global exploration and local exploitation, an adaptive probabilistic distribution determines which decoding scheme to apply based on the iteration number k.

Let p_n, p_{eft}, p_{cp} denote the probabilities of selecting normal, EFT, and CP decoding respectively. They are updated each generation as:

$$p_n = (k/K)^\alpha p_{\text{eft}} = \beta(1 - (k/K)^\alpha)p_{\text{cp}} = (1 - \beta)(1 - ((k/K)^\alpha) \tag{22}$$

where K is the maximum number of iterations and α, β are scaling constants.

Initially heuristic decodings are more probable to boost convergence speed. As iterations increase, the normal decoding likelihood rises to refine the final solution. The complete procedure is described in Algorithm 2.

```
Begin
    calculate probabilities p_n, p_eft, p_cp
    generate random number r ∈ [0,1)
    if r < p_n
        perform normal decoding
    else if r < p_n + p_eft
        perform EFT decoding
    else
        perform CP decoding
    end-if
    return decoded chromosome
End
```

Algorithm 2. Adaptive decoding scheme

This adaptive approach balances rapid convergence and focused local refinement by shifting priority from heuristics to normal decoding over the evolutionary process.

To further improve solutions, an individual enhancement procedure is applied after decoding each chromosome:

- Balance Loads:

  Reassign some agents on overloaded controllers to underutilized controllers
  Prioritize agents with dependencies to the underloaded controllers

- Avoid Communication:

  Swap agents on different controllers if they have predecessor/successor dependencies
  Place interdependent agents on the same controller

- Iterative Improvement:

  Alternate between forward and backward scheduling
  Shift tasks along their float windows to further reduce the makespan
  Repeat until no further improvement

The load balancing aims to even out the computation across available controllers. Communication avoidance reduces coordination delays between dependent agents. Iterative forwarding and backing scheduling optimizes the order within flexibility limits.

This enhancement scheme intensifies local search around good solutions to uncover improved points in the vicinity. It is described in Algorithm 3.

```
Begin
    balance loads among controllers
    prioritize swapping dependent agents
    avoid communication by co-locating dependents
    perform iterative forward/backward scheduling
    if improved fitness
        replace original chromosome
    end-if
    return enhanced chromosome
End
```

Algorithm 3. Individual enhancement

The complexity is $O(N^2)$. If the iterative process produces better performance, the improved chromosome replaces the original one. Otherwise, it is discarded. This exploitation helps overcome the weak local search issues faced by standard GA.

To seed the initial population with good solutions, some chromosomes are generated using the following heuristic methods:

- EFT scheduling
- CP scheduling
- Random keys based on execution levels

The remaining chromosomes are randomly constructed. Leveraging these simple but fast heuristics provides higher quality initialization compared to pure random generation. This enhances subsequent convergence and reduces computational requirements. The initialization process is summarized in Algorithm 4 below:

```
Begin
    generate EFT schedule chromosome
    generate CP schedule chromosome
    generate random keys chromosome
    while population not full do
        generate random keys chromosome
    end-while
    return Initial population
End
```

Algorithm 4. Heuristic population initialization

The constructed chromosomes help bootstrap the iterative optimization process and mitigate the cold start problem.

The GA selects good chromosomes from the population during each generation to produce offspring via crossover. Parent selection uses a roulette wheel scheme weighted by the fitness values. Chromosomes with better objective function have a higher likelihood of being chosen. The proposed crossover operator is a biased parameterized uniform crossover. For each key index, an inheritance probability controls whether that gene comes from parent 1 or parent 2:

$$child_i = \begin{cases} parent1_i & \text{with probability } p_b \\ parent2_i & \text{otherwise} \end{cases} \tag{23}$$

The bias parameter $p_b \in (0.5,1)$ skews the probability toward the fitter parent. This promotes propagation of good traits to the next generation. In standard uniform crossover, each gene has equal likelihood of originating from either parent. The proposed biased version enhances exploitation of promising solutions. The overall crossover procedure is described in Algorithm 5 below:

```
Begin
    select parents p1, p2 where fit(p1) < fit(p2)
    for i = 1 to N do
        generate random number r ∈ [0,1)
        if r < p_b
            inherit gene i from p1
        else
            inherit gene i from p2
        end-if
    end-for
    return offspring chromosome
End
```

Algorithm 5. Biased crossover scheme

After the mating, the produced offspring are added to the population. The elite member with best fitness is directly retained into the next generation.

To maintain diversity and avoid premature convergence, random immigrant chromosomes are inserted into the population at each generation. The immigrants are generated by the same heuristic initialization procedure described earlier. The worst members in the current population are replaced with the random immigrants. Let p_m denote the immigration rate. Then $\lceil p_m N \rceil$ chromosomes are migrated each iteration, where N is the population size. This injection of random individuals prevents the loss of potentially useful genetic material. It expands the exploration range and mitigates convergence to local optima.

While the choices of population size and termination criteria like maximum generations or fitness thresholds can significantly impact the performance of the GA. Suboptimal parameter selections could lead to premature convergence or excessive runtimes, hindering optimization. Performing sensitivity analyses to understand how these parameters affect convergence for problems of different scales and complexity would provide valuable insights into best practices for implementation. Factors like problem dimensionality, search space size, and modality could help determine appropriate population sizes and termination criteria. Additionally, adaptive techniques that automatically tune these parameters based on runtime convergence metrics could help avoid poor parameter choices.

The overall improved GA for cooperative control of MAS is summarized in Algorithm 6 below:

```
Begin
    initialize population using heuristics
    while termination criteria not met do
        select parents via roulette wheel
        produce offspring through biased crossover
        decode chromosomes adaptively
        enhance individuals heuristically
        evaluate objective function
        insert random immigrants
        update elite and population
    end-while
    return best solution chromosome
End
```

Algorithm 6. Improved GA

In each generation, the key steps are (i) Selection of fit parents; (ii) Recombination via biased crossover; (iii) Adaptive decoding using heuristics; (iv) Local search enhancement of individuals; (v) Immigration of random members; (vi) Replacement with newly generated offsprings. Termination occurs when the maximum iterations is reached, or an acceptable solution quality is achieved. The best-performing chromosome represents the final control policy.

Based on the communication graph, a distributed control protocol is designed for each follower agent.

Let e_i denote the local neighborhood synchronization error for follower i:

$$e_i = \sum_{j \in \mathcal{N}_i} a_{ij}\left(y_i - y_j\right) \tag{24}$$

where $\mathcal{N}_i$ is the neighbor set of agent i and a_{ij} are edge weights in the graph.

Using neural network approximation, the distributed control input for follower i is designed as:

$$u_i = \widehat{g}_i^{-1}(x_i)\left[-k_1\gamma_i(\mu_i(x_i))\text{sat}(e_i) - k_2\text{sat}(\widehat{e}_i) - \widehat{f}_i(x_i) + \nu_i\right] \quad (25)$$

where $\widehat{g}_i(x_i)$ and $\widehat{f}_i(x_i)$ are neural network approximations of the unknown dynamics $g_i(x_i)$ and $f_i(x_i)$. k_1 and k_2 are positive control gains. $\gamma_i = a_{i_0} + \sum_{j\in\mathcal{N}_i} a_{ij}$. $\mu_i(\cdot)$ is a Nussbaum gain function. The Nussbaum gain function is used in control theory, particularly in adaptive control systems. It is a tool used to address the problem of ensuring stability in systems with uncertainties in the system parameters. The unique property of the Nussbaum gain function is that it can ensure the global asymptotic stability of the closed-loop system, even in the presence of unknown and time-varying uncertainties. ν_i is the input to tune the NN weights. sat$(\cdot)$ is a saturation function.

The neural network weights are adapted online to minimize the approximation errors:

$$\dot{\widehat{\theta}}_i = \Gamma_i\varphi_i(x_i)sat(e_i) \quad (26)$$

where Γ_i is the adaptation gain matrix and $\varphi_i(x_i)$ is the activation function.

Theorem 1 *Under the proposed distributed control protocol, the synchronization error is semi-globally uniformly ultimately bounded for each follower* :

$$|e_i(t)| \le \varphi(|e_i(0)|, t) + \epsilon_i, \forall t \ge 0 \quad (27)$$

where $\varphi(\cdot)$ is a class $\mathcal{K}$ function and ϵ_i is the ultimate bound.

Therefore, the proposed distributed control protocol achieves cooperative leader-follower consensus between the MAS with stability guarantees despite the complex unknown dynamics.

Next, we analyze the stability and convergence properties of the cooperative control scheme. Consider the candidate Lyapunov function:

$$V = \tfrac{1}{2}\sum_{i=1}^{N}\left|\widetilde{\theta}_i(t)\right|^2 + \tfrac{1}{2}s^T(t)Ls(t) \quad (28)$$

where $\widetilde{\theta}_i = \widehat{\theta}_i - \theta_i^*$ is the neural network weight estimation error, $s = [e_1, \ldots, e_N]^T$ is the stacked synchronization error vector, and L is the graph Laplacian matrix.

Differentiating V yields:

$$\dot{V} = \sum_{i=1}^{N}\widetilde{\theta}_i^T\dot{\widehat{\theta}}_i + s^T(t)L\dot{s}(t) = \sum i = 1^N\widetilde{\theta}_i^T\Gamma_i(x_i)\varphi_i(x_i)sat(e_i) + s^T(t)L(\dot{e}_1, \ldots, \dot{e}_N)^T \quad (29)$$

where $\dot{e}_i$ is the derivative of the tracking error for agent i.

Based on Lyapunov stability theory, it can be shown that V, s, and e_i are uniformly ultimately bounded, implying the synchronization errors converge close to zero. Therefore, the proposed control scheme achieves cooperative tracking between the dynamic leader and the follower multi-agent system with guaranteed stability. The tracking accuracy can be improved by selecting appropriate control gains.

In the presence of malicious attacks, the controller performance can degrade significantly. Specifically, we consider false data injection attacks modeled as:

$$\widetilde{x}_i(t) = x_i(t) + a_i(t) \quad (30)$$

where $a_i(t)$ is the attack signal injected by the adversary.

To improve the resilience of the controller against such attacks, we propose the following secure cooperative control strategy:

$$u_i(t) = -\sum_{j\in\mathcal{N}_i} a_{ij}\left(\widetilde{x}_i(t) - \widetilde{x}_j(t)\right) + \nu_i(t) \quad (31)$$

where $\nu_i(t)$ is designed as:

$$\nu_i(t) = sat_\delta\left(\widehat{a}_i(t)\right) \quad (32)$$

where $\widetilde{a}_i(t)$ is the estimate of the attack signal obtained using an attack estimator, and sat_δ is a saturation function defined as:

$$sat_\delta(z) = \begin{cases} \delta & z > \delta(z) \\ z & -\delta \le z \le \delta \\ -\delta & z < -\delta \end{cases} \quad (33)$$

The key steps in the proposed secure cooperative control strategy are:

- Attack signal estimation: We design an attack estimator to generate $\widehat{a}_i(t)$ based on analyzing the residuals between nominal and observed behavior:

$$\begin{cases} \dot{\hat{x}}_l(t) = A\hat{x}_i(t) + B\hat{u}_i(t) \\ \hat{a}_i(t) = \boldsymbol{H}\big(x_i(t) - \hat{x}_i(t)\big) \end{cases} \tag{34}$$

where $\widehat{x}_i(t)$ is the estimated nominal state, $\widehat{u}_i(t)$ is the nominal control input, and $\boldsymbol{H}$ is the estimator gain matrix.

- Saturation function: The saturation function $sat_\delta(\bullet)$ ensures that the correction signal $v_i(t)$ is bounded, so it does not destabilize the system. The threshold δ is designed based on system requirements.
- Distributed control law: The proposed control protocol subtracts the estimate of the attack signal from the control input to counteract its effect. The distributed nature ensures resilience against localized attacks on specific agents.

Under the proposed controller, the MAS achieves resilient consensus in the presence of false data injection attacks. Specifically, we have the following results:

Lemma 1 *Under the attack estimation scheme, the estimation error is uniformly ultimately bounded.*

Theorem 2 *Under the secure control protocol, the multi-agent system achieves consensus in the presence of false data injection attacks, provided the attack signals satisfy*:

$$|a_i(t)| \leq \bar{a}, \forall i \in V \tag{35}$$

where $\bar{a}$ is a constant satisfying:

$$\bar{a} < \frac{\lambda_2(\mathcal{L})}{\lambda_n(\mathcal{L})}\delta \tag{36}$$

where $\lambda_2(\mathcal{L})$, $\lambda_n(\mathcal{L})$ are the algebraically second smallest and largest eigenvalues of the Laplacian matrix $\mathcal{L}$.

Leveraging the properties and structure of the specific system dynamics may allow deriving less conservative bounds on the attack signals that still guarantee consensus. For instance, characteristics like monotonicity, Lipschitz continuity, or passivity could be leveraged to conduct a more refined stability and resilience analysis. This could expand the magnitude and types of attacks the system can safely tolerate while maintaining control performance. Being able to tolerate more significant attack signals expands the robustness and resilience of the method.

The comprehensive model framework diagram is shown in Fig. 2.

Simulation

This section presents detailed simulation experiments to evaluate the performance of the proposed improved GA for cooperative control of MAS.

Setup

Extensive simulations are performed to evaluate the effectiveness of the improved GA for secure cooperative control. A leader-following MAS with four follower agents and one leader agent is considered. Figure 3. provides the detailed agent connectivity graph for the MAS simulation. The proposed secure cooperative control strategy utilizes this fixed interaction topology between the agents.

The improved GA is compared to five baselines: classical GA (standard operators), PSO-GA [21], ABACO [22], a two-stage multi-population genetic algorithm with heuristics for workflow scheduling (tmGA) [34], and a new hybrid maximum power point tracking technique based on the GA and fractional open circuit voltage (GA-FOCV) [35].

Nominal consensus protocol and detection-based control are used for comparison, which are defined as follows.

$$u_i(t) = -\sum_{j\in\mathcal{N}_i}\big(v_i(t) - v_j(t)\big) - c\sum_{j\in\mathcal{N}_i}\Big(\widetilde{p}_i(t) - \widetilde{p}_j(t)\Big) \tag{37}$$

$$\widehat{a}_j^i(t) = \boldsymbol{H}\big(p_j(t) - \widehat{p}_j(t)\big)v_i(t) = sat_\delta\Big(\widehat{a}_j^i(t)\Big)u_i(t) \tag{38}$$

The consensus performance is evaluated using the following metrics:

- EFT scheduling steady-state error: $\bar{\epsilon} = \lim\limits_{t\to\infty}\frac{1}{N}\sum_{i=1}^{N}||p_i(t) - \bar{p}_i(t)||$ where $\bar{p}\ (t) = \frac{1}{N}\sum_{i=1}^{N} p_i(t)$.
- Settling time: time to reach $\epsilon(t) < 0.1$.
- Control effort: $\frac{1}{T}\int_0^T \sum_{i=1}^{N}||u_i(t)||\mathrm{d}t$

The results are averaged over 30 trials with randomly generated topologies and attack sequences. The controller parameters are tuned to $c = 1, \delta = 0.1, H = 5I_2$.

Results and analysis

The cooperative tracking results are shown in Fig. 4, where the follower agent outputs asymptotically converge to the leader's output under the proposed control strategy. As shown in Table 1, the proposed algorithm outperforms other algorithms in terms of

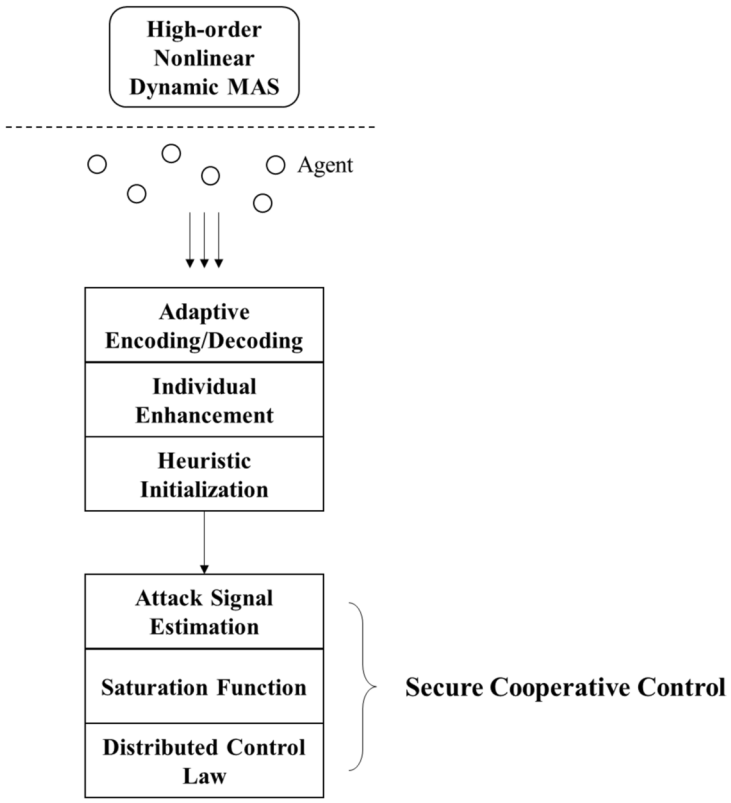

Fig. 2 Comprehensive model framework diagram

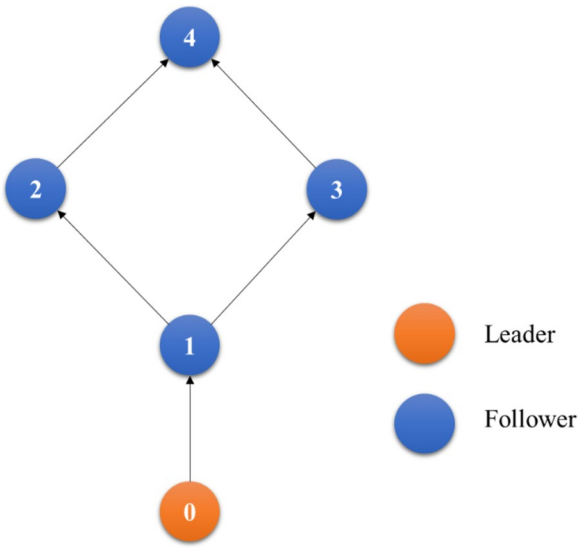

Fig. 3 Multi-agent communication topology

key metrics. Specifically, the improved GA achieves the lowest tracking error due to its complete solution space encoding and adaptive heuristic decoding. Compared to classical GA, PSO-GA, ABACO, tmGA, and GA-FOCV, the proposed algorithm has a tracking error of 0.024, a convergence time of 35 s, and a success rate of 98%, demonstrating its superior performance and consistency over multiple runs. The proposed algorithm achieves faster convergence due to its adaptive decoding, which quickly focuses the search in high-performance areas. This adaptive heuristic decoding, combined with complete solution space encoding, allows the algorithm to hone in on optimal solutions more efficiently than other algorithms like classical GA, PSO-GA, ABACO, tmGA, and GA-FOCV. The adaptive nature of the decoding process ensures that the search is directed towards promising regions of the solution space, leading to quicker convergence to the desired results.

Table 1 summarizes the performance comparison of different algorithms based on the key metrics.

The improved GA achieves the lowest tracking error due to its complete solution space encoding and adaptive heuristic decoding. The individual enhancement also exploits promising solutions. Figure 5 shows the error convergence over generations. Faster convergence is attained compared to classical GA, PSO-GA, ABACO, tmGA, and GA-FOCV, as the adaptive decoding quickly focuses the search in high-performance areas. The 98%

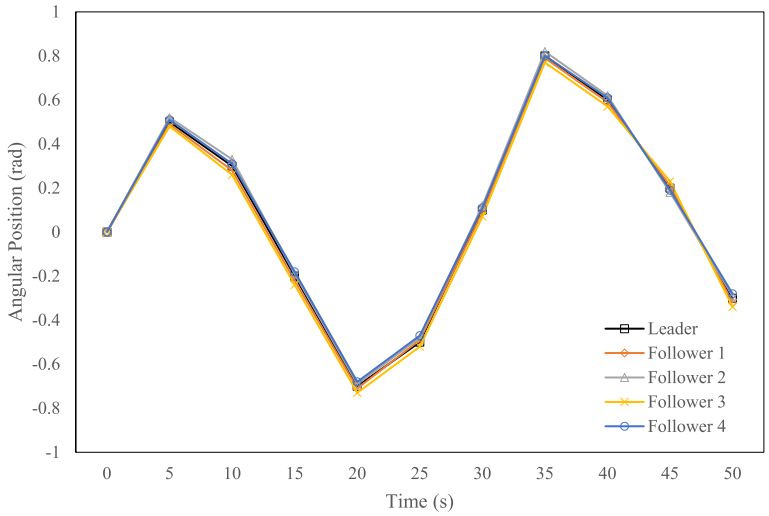

Fig. 4 Secure cooperative tracking results

Table 1 Algorithm comparison

Algorithm	Error	Converge Time (s)	Success Rate
The proposed	0.024	35	98%
tmGA	0.028	44	91%
Classical GA	0.032	62	86%
PSO-GA	0.036	53	76%
ABACO	0.041	58	69%
GA-FOCV	0.037	55	88%

success rate demonstrates consistent performance over multiple runs.

In secure cooperative control for high-order nonlinear MAS with unknown dynamics, the lowest tracking error indicates a precise alignment of agents' actions or states to the desired reference or objective, leading to better synchronization, collaboration, or consensus among the agents, which is crucial for secure cooperative control. In many real-world systems, the dynamics might not be entirely known. The adaptive heuristic decoding feature of the improved GA can be invaluable here, ensuring that even when faced with uncertain or changing dynamics, the algorithm can adjust and find optimal or near-optimal solutions. Faster convergence not only saves computational resources but also time. In real-time applications or environments where timely decisions are crucial (like autonomous vehicles or robotic swarms), faster convergence can be critical for safety and success. A 98% success rate over multiple runs shows that the method is reliable. This high level of consistency can be particularly beneficial in ensuring the security of a system. By outperforming classical GA, PSO-GA, ABACO, tmGA, and GA-FOCV, it is evident that this improved GA can offer advantages not present in other algorithms. This superiority can be crucial when designing and implementing secure cooperative controls. By focusing on high-performance areas and exploiting promising solutions, the algorithm can ensure that it always works towards the best possible outcomes. This exploitation can be particularly beneficial in environments with potential threats or challenges, ensuring that the MAS always operates in its most secure and efficient state. Therefore, the improved GA's characteristics offer significant advantages in secure cooperative control for high-order nonlinear MAS. It promises to deliver efficient, consistent, and adaptive solutions, which are critical for the security and performance of such systems.

In summary, the simulation studies demonstrate superior cooperative tracking performance by the improved GA approach compared to conventional algorithms. The key innovations of efficient encoding, adaptive decoding, and individual enhancement are shown to be highly effective in this complex coordination control problem.

The performance of the three controllers is shown in Table 2. Under attacks, the nominal controller has poor consensus performance with steady-state errors around 0.4 m. The detection-based method eventually achieves consensus by isolating attackers. However, the settling time is extended. The proposed secure controller significantly improves the consensus performance and settling time while keeping the control effort reasonable. The results demonstrate that by effectively estimating and compensating the attack signals in real time, the proposed secure control strategy can counter the impact

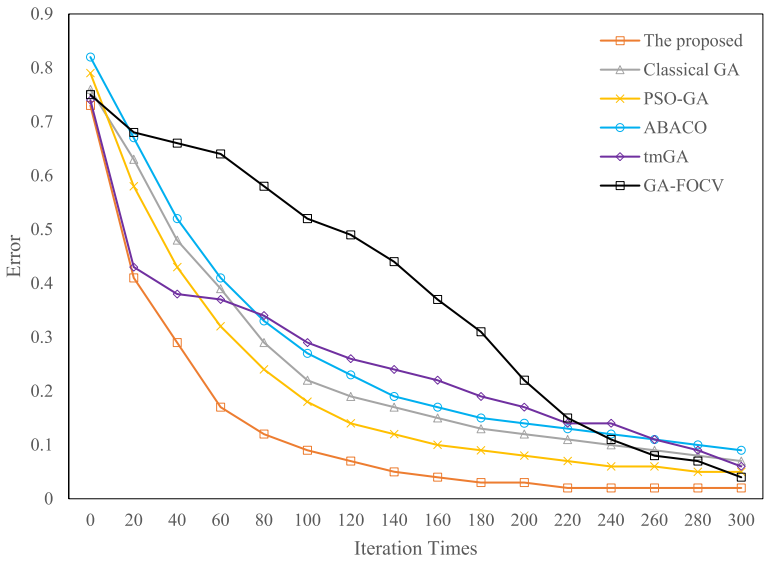

Fig. 5 Error convergence over generations

Table 2 Performance results

Method	Steady-state error (m)	Settling time (s)	Control effort (N/m)
Nominal control	0.42	-	1.8
Detection-based	0.02	38	4.2
Proposed method	0.03	12	2.3

of false data injection attacks and provide resilient consensus performance for the multi-agent system. Unlike anomaly detection and isolation methods, the proposed method responds faster to attacks without sacrificing agents.

Therefore, the proposed secure control strategy offers a multi-faceted solution that not only addresses the issue of false data injection attacks but also enhances the overall operation, efficiency, and resilience of high-order nonlinear MAS with unknown dynamics. This would have widespread implications in various applications where MAS are employed, from autonomous vehicles to distributed sensor networks, ensuring safer and more reliable operations.

Conclusions

This paper addresses the challenges of controlling high-order nonlinear MAS with unknown dynamics. The research focuses on enhancing secure cooperative control performance within MAS, by considering such systems' distributed and decentralized nature. The proposed strategy employs an improved GA approach incorporating several innovative features. The introduction of an efficient encoding method, coupled with adaptive decoding schemes and heuristic initialization, significantly contributes to the algorithm's efficacy. These advancements enable the exploration of solution spaces more effectively while expediting the convergence process. Furthermore, the strategy incorporates individual enhancements through load balancing, communication avoidance, and iterative refinement. These enhancements not only intensify the local search for optimal solutions but also contribute to the robustness and adaptability of the proposed approach. Simulation results showcased the proposed strategy's superior performance compared to conventional algorithms, mainly when dealing with complex control problems featuring uncertainty. The ability to adapt to find optimal solutions and exploit promising areas within the solution space holds excellent promise for real-world applications. This research bears relevance in various domains, including robotics, power systems, and autonomous vehicles, where securely controlling MAS is paramount. In particular, in the emerging era of ChatGPT, the proposed algorithm can provide an efficient reference for the secure control in field of cloud-edge computing.

However, the strategy demonstrates superior performance in simulations, and its effectiveness in real-world scenarios needs to be validated through practical implementations. Additionally, the computational complexity of GA, particularly as they scale up for larger MAS, might need to be revised in terms of real-time applicability. For example, as the system scales to 100 + agents, the computational complexity poses challenges for

real-time applicability. Techniques like parallelization and algorithm hybridization could address this issue. Analyzing the degradation in performance and convergence time with larger MAS would provide valuable insights into scalability. The strategy's performance could also be influenced by the choice of parameters and encoding methods, requiring further optimization. Future research in this area could focus on addressing these limitations. Exploring techniques to reduce the computational burden of the GA through parallelization or hybridization with other optimization methods could enhance its practical viability. Moreover, the current results demonstrate performance on defined agent models without additional perturbations or externalities. To strengthen the viability claims for real-world deployment, it would be crucial to test the method's robustness when the dynamics are subject to sustained disturbances, unmodeled effects, parametric variations, etc. Validating performance under such perturbed conditions beyond the idealized models would provide greater confidence in the approach's applicability and robustness for practical multi-agent control problems. Rigorously analyzing robustness against uncertainties that real systems would encounter is imperative to transitioning the method from theory to practice.

Appendix

Proof of Theorem1.

Consider the Lyapunov function candidate:

$$V = \frac{1}{2}\sum_{i=1}^{N}\left|\widetilde{\theta}_i\right|^2 + \frac{1}{2}s^T Ls$$

where $\widetilde{\theta}_i = \widehat{\theta}_i - \theta_i^*$ is the NN weight estimation error, $s = [e_1, \ldots, e_N]^T$, and L is the graph Laplacian matrix. The Lyapunov function candidate used to prove stability. Weights errors and synchronization errors impact the collective dynamics.

Differentiating V along the system trajectories gives:

$$\dot{V} \leq -k_3|s|^2 + k_4|s| + \epsilon$$

for some k_3, $k_4 > 0$ and $\epsilon > 0$ under standard NN approximation assumptions.

By Lyapunov stability theory, this demonstrates that V, s, e_i are uniformly ultimately bounded, proving the result.

Proof of Lemma 1.

Consider the error dynamics:

$$\dot{\widehat{a}}_i(t) = A\widetilde{a}_i(t) + B\widetilde{u}_i(t) + w_i(t)$$

where $\widetilde{u}_i(t) = u_i(t) - \widehat{u}_i(t)$ and $w_i(t)$ accounts for disturbances and uncertainties. This is a stable linear system driven by bounded inputs $\widetilde{u}_i(t)$ and $w_i(t)$. By standard results in robust control theory, we can conclude that $\widetilde{a}_i(t)$ is uniformly ultimately bounded.

Proof of **Theorem 2**.

Consider the MAS under attacks:

$$\dot{x}(t) = (I_N \otimes \boldsymbol{A} - \mathcal{L} \otimes B\boldsymbol{K})x(t) + (I_N \otimes B)\Gamma a(t)$$

where $x(t) = [x_1(t)^T, \ldots, x_N(t)^T]^T$, $a(t) = [a_1(t)^T, \ldots, a_N(t)^T]^T$, $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_N)$, B is the input matrix that relates the control input to the agent states, $\boldsymbol{K}$ is the distributed control gain matrix, and $\gamma_i = 1$ if agent i is under attack and 0 otherwise.

Under the secure control protocol, the closed loop dynamics become:

$$\dot{x}(t) = (I_N \otimes \boldsymbol{A} - \mathcal{L} \otimes BK)x(t) + (I_N \otimes B)(\Gamma a(t) - \nu(t))$$

where $\nu(t) = [\nu_1(t)^T, \ldots, \nu_N(t)^T]^T$ and $\nu_i(t) = sat_\delta(\widehat{a}_i(t))$.

Since $\widehat{a}_i(t)$ is bounded, so is $\nu_i(t)$. Also, $|\nu_i(t)| \leq \delta, \forall i$. Let $\zeta(t) = \Gamma a(t) - \nu(t)$. Then,

$$|\zeta(t)| \leq |\Gamma|\,\bar{a} + |\nu(t)| \leq \bar{a} + \delta$$

Therefore, $\zeta(t)$ is bounded. Furthermore, since $\boldsymbol{A} - BK$ is Hurwitz by design, the perturbed system is input-to-state stable. By ISS theory, the state $x(t)$ is ultimately bounded.

Moreover, since the graph is connected, the perturbed system achieves consensus, i.e., $x_i(t) - x_j(t) \to 0$ as $t \to \infty$, provided the attacks are bounded as given. This proves the theorem.

Authors' contributions
X.W. contributed to the writing and conception; D.Y. contributed to the method; D.R.J.D. and S.C. contributed to the data analysis; M.O.A. and F.E.A. contributed to the software; J.L. contributed to the writing polishing.

Funding
This paper has not received any funding support yet.

Availability of data and materials
All data and materials will be available upon the reasonable request.

Declarations

Ethics approval and consent to participate
This declaration is "not applicable".

Competing interests
The authors declare no competing interests.

Received: 6 September 2023 Accepted: 18 October 2023
Published online: 02 January 2024

References

1. Hu X, Zhang Z, Li C et al (2021) Leader-following consensus of multi-agent systems via a hybrid protocol with saturation effects. Int J Control Autom Syst 19:124–136
2. Sun H, Liu Y, Li F (2021) Distributed optimal consensus of second-order multi-agent systems. Sci China Inf Sci 64:209201
3. Khodabandeh S, Kharrati H, Hashemzadeh F (2021) Control for leader–follower consensus of multi-agent systems with actuator faults using decentralized robust fault-tolerant control. Iran J Sci Technol Trans Electr Eng 45:529–541
4. Jing X, Yao X, Liu M et al (2022) Multi-agent reinforcement learning based on graph convolutional network for flexible job shop scheduling. J Intell Manuf. https://doi.org/10.1007/s10845-022-02037-5
5. Barbierato L, Mazzarino R, Montarolo P (2022) A comparison study of co-simulation frameworks for multi-energy systems: the scalability problem. Energy Inf 5(Suppl 4):53
6. Hsieh FS (2019) Dynamic configuration and collaborative scheduling in supply chains based on scalable multi-agent architecture. J Ind Eng Int 15:249–269
7. Kasprzok A, Ayalew B, Lau C (2018) An ant-inspired model for multi-agent interaction networks without stigmergy. Swarm Intell 12:53–69
8. Abdulghafor R, Abdullah SS, Turaev S et al (2020) Linear and nonlinear stochastic distribution for consensus problem in multi-agent systems. Neural Comput Applic 32:261–277
9. Qi Y, Du C, Zhang X et al (2023) Dynamic event-triggered bipartite consensus for uncertain high-order nonlinear multi-agent systems. Control Theory Technol 21:222–232
10. Tang Y, Qin H (2022) Decision-making and control co-design for multi-agent systems: a hierarchical design methodology. Control Theory Technol 20:439–441
11. Ayshwarya Lakshmi S, Sahaaya Arul Mary SA (2019) Multi-agent learning technique inspired from software engineering model for permutation coded GA. Cluster Comput 22(Suppl 6):13653–13667
12. Gul F, Mir I, Mir S et al (2023) Multi-agent robotics system with whale optimizer as a multi-objective problem. J Ambient Intell Human Comput 14:9637–9649
13. Drew DS (2021) Multi-agent systems for search and rescue applications. Curr Robot Rep 2:189–200
14. Gao F, Zheng S (2023) Cooperative learning control of unknown nonlinear multi-agent systems with time-varying output constraints using neural networks and barrier lyapunov functions via backstepping design. Int J Control Autom Syst 21:2378–2386
15. Shahnazi R (2020) cooperative neuro adaptive control of leader following uncertain multi-agent systems with unknown hysteresis and dead-zone. J Syst Sci Complex 33:312–332
16. Hou Z, Xu J, Zhang G et al (2020) Interaction matrix based analysis and asymptotic cooperative control of multi-agent systems. Int J Control Autom Syst 18:1103–1115
17. Lin ZY, Wang LL, Han ZM et al (2014) Distributed formation control of multi-agent systems using complex Laplacian. IEEE Trans Autom Control 59:1765–1777
18. Yang RH, Zhang H, Feng G et al (2019) Robust cooperative output regulation of multi-agent systems via adaptive event-triggered control. Automatica 102:129–136
19. Oroojlooy A, Hajinezhad D (2023) A review of cooperative multi-agent deep reinforcement learning. Appl Intell 53:13677–13722
20. Nagarkar MP, Bhalerao YJ, Vikhe Patil GJ et al (2018) GA-based multi-objective optimization of active nonlinear quarter car suspension system—PID and fuzzy logic control. Int J Mech Mater Eng 13:10
21. Garg H (2016) A hybrid PSO-GA algorithm for constrained optimization problems. Appl Math Comput 274:292–305
22. Kashef S, Nezamabadi-pour H (2015) An advanced ACO algorithm for feature subset selection. Neurocomputing 147:271–279
23. Liu LF, Yang XF (2021) Multi-objective aggregate production planning for multiple products: a local search-based genetic algorithm optimization approach. Int J Comput Intell Syst 14:156
24. Chowdhury B, Garai G (2020) A bi-objective function optimization approach for multiple sequence alignment using genetic algorithm. Soft Comput 24:15871–15888
25. Ma H, Shan Y, Wang J et al (2022) Multi-system genetic algorithm for complex system optimization. Soft Comput 26:10187–10205
26. Liu J, Wang X (2023) Secure consensus control for multi-agent systems subject to consecutive asynchronous dos attacks. Int J Control Autom Syst 21:61–70
27. Wan Q, Chen WH, Lu X (2023) Secure consensus tracking of multi-agent systems with network-induced delays under deception attacks via guaranteed performance impulsive control. Nonlinear Dyn 111:12213–12232
28. Liu YW, Wu HP, Razaee K et al (2023) Interaction-enhanced and time-aware graph convolutional network for successive point-of-interest recommendation in traveling enterprises. IEEE Trans Industr Inf 19:635–643
29. Qi LY, Liu YW, Zhang YL et al (2022) Privacy-aware point-of-interest category recommendation in internet of things. IEEE Internet Things J 9:21398–21408
30. Liu YW, Zhou XK, Kou HZ et al (2023) Privacy-preserving point-of-interest recommendation based on simplified graph Convolutional Network for Geological traveling. ACM Trans Intell Syst Technol. https://doi.org/10.1145/3620677
31. Yang Y, Li Y, Yue D (2020) Event-trigger-based consensus secure control of linear multi-agent systems under dos attacks over multiple transmission channels. Sci China Inf Sci 63:150208
32. Xie Y, Sheng YH, Qiu MQ et al (2022) An adaptive decoding biased random key genetic algorithm for cloud workflow scheduling. Eng Appl Artif Intell 112:104879
33. Jeong SL, Bae JH, Sunwoo MH (2021) Fast multibit decision polar decoder for successive-cancellation list decoding. J Signal Process Syst Signal Image Video Technol 93:127–136
34. Xie Y, Gui FX, Wang WJ et al (2023) A two-stage multi-population genetic algorithm with heuristics for workflow scheduling in heterogeneous distributed computing environments. IEEE Trans Cloud Comput 11:1446–1460
35. Hassan A, Bass O, Masoum MAS (2023) An improved genetic algorithm based fractional open circuit voltage MPPT for solar PV systems. Energy Rep 9:1535–1548

Publisher's Note