mathematics

MDPI

Article

Optimization of Predefined-Time Agent-Scheduling Strategy Based on PPO

Dingding Qi [1], Yingjun Zhao [1], Longyue Li [1,*] and Zhanxiao Jia [2]

[1] Air Defense and AntiMissile School, Air Force Engineering University, Xi'an 710043, China; tintin0711@163.com (D.Q.); zyj6681@163.com (Y.Z.)
[2] Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China; jiazhanxiao2022@163.com
* Correspondence: lilong_yue@hotmail.com

Abstract: In this paper, we introduce an agent rescue scheduling approach grounded in proximal policy optimization, coupled with a singularity-free predefined-time control strategy. The primary objective of this methodology is to bolster the efficiency and precision of rescue missions. Firstly, we have designed an evaluation function closely related to the average flying distance of agents, which provides a quantitative benchmark for assessing different scheduling schemes and assists in optimizing the allocation of rescue resources. Secondly, we have developed a scheduling strategy optimization method using the Proximal Policy Optimization (PPO) algorithm. This method can automatically learn and adjust scheduling strategies to adapt to complex rescue environments and varying task demands. The evaluation function provides crucial feedback signals for the PPO algorithm, ensuring that the algorithm can precisely adjust the scheduling strategies to achieve optimal results. Thirdly, aiming to attain stability and precision in agent navigation to designated positions, we formulate a singularity-free predefined-time fuzzy adaptive tracking control strategy. This approach dynamically modulates control parameters in reaction to external disturbances and uncertainties, thus ensuring the precise arrival of agents at their destinations within the predefined time. Finally, to substantiate the validity of our proposed approach, we crafted a simulation environment in Python 3.7, engaging in a comparative analysis between the PPO and the other optimization method, Deep Q-network (DQN), utilizing the variation in reward values as the benchmark for evaluation.

Keywords: PPO; scheduling strategy; predefined time; multi-agent systems; rescue dispatching

MSC: 90B36

check for updates

Citation: Qi, D.; Zhao, Y.; Li, L.; Jia, Z. Optimization of Predefined-Time Agent-Scheduling Strategy Based on PPO. *Mathematics* **2024**, *12*, 2387. https://doi.org/10.3390/math12152387

Academic Editor: Janez Žerovnik

Received: 21 June 2024
Revised: 23 July 2024
Accepted: 27 July 2024
Published: 31 July 2024

Copyright: © 2024 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

With the rapid development of technology, the application of agents in the field of rescue has become increasingly widespread [1–3]. However, how to efficiently dispatch agents for rescue missions and ensure their accurate and rapid arrival at designated locations remains a significant issue facing current research. Traditional agent-dispatching methods often lack precise considerations for real-time performance and average arrival time [4,5], while control strategies may experience performance degradation in complex environments [6]. Therefore, the study of new methods for agent rescue dispatching and control is of great significance for improving rescue efficiency and reducing rescue costs.

In the context of rescue operations, the promptness of agent arrival holds paramount importance in determining the overall effectiveness of the rescue efforts [7–9]. Men et al. [10] proposed a multi-objective emergency response facility location (ERFsL) model for emergency management, taking into account the spatial characteristics of facilities and the potential risks associated with catastrophic interlocking chemical accidents. Regarding the application of swarm intelligence algorithms in positioning, Shalli Rani et al. [11] proposed a method that utilizes the krill herd algorithm for the non-ranging distributed processing

of mobile target nodes in a maritime rescue network. Addressing the emergency dispatch issue for large-scale forest fires with multiple rescue centers (depots) and limited firefighting resources, Wang et al. [12] aim to minimize the total completion time of all firefighting tasks by determining the optimal rescue routes for firefighting teams from multiple rescue centers. Nonetheless, previous research has frequently neglected the need for the precise estimation and optimization of agent arrival times, thereby compromising the efficiency of scheduling strategies. By constructing an average distance function, we are able to more precisely evaluate the arrival time of agents under different scheduling strategies, thus optimizing the scheduling plan and ensuring that the agents arrive at the rescue site in the shortest possible time.

Amidst the complexity and volatility of rescue environments, a rational and swift dispatch strategy serves to optimize resource allocation and elevate utilization efficiency [13–15]. Traditional scheduling optimization methods often rely on mathematical programming techniques [16,17], such as linear programming, integer programming, and mixed-integer linear programming. These methods formulate scheduling problems as optimization problems with the objective of minimizing a specific cost function, such as makespan, tardiness, or total completion time. They utilize established algorithms, including simplex methods, branch-and-bound, and heuristic search algorithms, to find optimal or near-optimal solutions. Yue et al. [18] established an evaluation model for electric vehicle (EV) indicators, determined the weight of each indicator using the optimal combination weight method based on accelerated genetic algorithms, and proposed an optimized scheduling strategy to achieve optimal EV fleet scheduling. For the large-scale and time-consuming agile Earth observation satellite scheduling problem, Du et al. [19] propose a data-driven parallel scheduling method that incorporates a probability prediction model, a task allocation strategy, and a parallel scheduling approach. In the IEEE Time-Sensitive Networking (TSN) Task Group, Min et al. [20] proposed two metaheuristic optimization methods to enhance the scheduling success rate of the Time-Aware Shaper (TAS). However, these traditional methods can suffer from limitations, particularly when dealing with complex real-world scheduling problems characterized by uncertainty, dynamic changes, and non-linear constraints [21]. They may struggle to scale to large-scale instances, require significant computational resources, and often make assumptions that do not fully capture the complexities of real-world systems. Furthermore, traditional methods may not be flexible enough to accommodate emerging trends and requirements, such as real-time decision making and integration with machine learning techniques. To overcome the previously mentioned limitations, a scheduling strategy optimization method based on the PPO algorithm is designed. Combining the characteristics of the PPO algorithm, an optimization method suitable for agent rescue scheduling is designed, achieving the efficient optimization of the agent-scheduling strategy and improving rescue efficiency.

Predefined-time control can ensure that the intelligent agent arrives at the designated location within the pre-set time limit during the rescue process, thus ensuring the timeliness and effectiveness of the rescue operation [22–24]. To enable the direct preset of the upper bound of the settling time, ref. [25] first proposed a new concept called predefined-time stability. Predefined-time control can make the control system achieve stability before the user-preset time, which has aroused the research interest of scholars. Refs. [26,27] separately studied the predefined-time control design issues for first-order and second-order systems. In [28], a predefined-time leader–follower consensus protocol was established to achieve the predefined-time convergence of tracking errors. The authors of [29] investigated the predefined-time bipartite consensus control problem for multiple quadrotors under malicious attacks. However, the aforementioned predefined-time control design methods are proposed for control systems that satisfy the matching conditions and cannot be used to address the problem for non-strict feedback nonlinear systems. To overcome the aforementioned constraints, we formulate a precision-oriented fuzzy adaptive tracking control approach with non-singular predefined time, specifically tailored for nonlinear systems. This approach synergistically leverages the strengths of fuzzy control and adaptive control,

empowering the agent with the capability to precisely calibrate control parameters during its dynamic motion, thereby effectively navigating through the complexities of diverse environmental scenarios.

The core innovations are highlighted as follows:

(1) In optimizing the dispatch of rescue agents, we have formulated an evaluation function centered around the average flying distance of the agents. This function serves as a rigorous assessment criterion for rescue mission scheduling, accurately quantifying the time taken by agents to reach their intended destinations.

(2) In the research on agent-task allocation, this paper designs a task allocation method based on the proximal policy optimization algorithm. This algorithm can adaptively adjust the task allocation strategy, thereby enhancing the collaborative efficiency of the agent team and improving the quality of task completion.

(3) Addressing the issue of controlling the arrival of agent at target locations, this paper proposes a singularity-free predefined-time fuzzy adaptive tracking control strategy tailored for nonlinear systems. The agents are capable of accurately reaching their destinations from the starting points with enhanced efficiency within a predefined time.

2. Preliminaries and Problem Description

2.1. Preliminaries

The following nonlinear system is to be taken into account:

$$\dot{x}(t) = f(x,t), x(0) = x_0, \tag{1}$$

where $x \in R^n$ signifies the state vector of the system and $f(x,t) : R^n \to R^n$ is designated as a continuous function.

Definition 1. *When the state trajectory $x(t,x_0)$ of system (1) fulfills the condition $\|x(x_0,t)\| \leq \varepsilon$ after a predetermined time T_d has elapsed, the equilibrium point is referred to as practical predefined-time stable (PPTS), where ε is a positive constant.*

Lemma 1 (See [30])**.** *Given a Lyapunov function $V(x)$ that pertains to system (1) and satisfies the inequality*

$$\dot{V} \leq -\frac{\pi}{\theta T_d} V^{1+\frac{\theta}{2}} - \frac{\bar{\rho}\pi}{\theta T_d} V^{1-\frac{\theta}{2}} + \Theta, \tag{2}$$

where T_d is the predefined time and is positive, θ and $\bar{\rho}$ are design parameters with $0 < \theta < 1$ and $\bar{\rho} = \sqrt{2}+1$, and D is a positive constant. Under these circumstances, the equilibrium point of system (1) is deemed PPTS with the predefined time T_d. Additionally, the residual set of the state trajectory $x(t,x_0)$ can be defined as

$$x \in \{V(x) : V(x) \leq 2\theta T_d \Theta/\pi\}. \tag{3}$$

2.2. Problem Description

In this paper, we are devoted to exploring the pivotal question of how to allocate resources and satisfy demands in rescue operations. Precisely, our operation involves the utilization of n rescue agents, all of which possess the capability to transport rescue packages with a standardized load capacity of m units. Furthermore, there are p disaster-affected rescue locations, each exhibiting varying degrees of need for rescue resources, which we quantify as q_i, where $i = 1, 2, ..., n$ signifies the individual rescue location. Concerning the issue of agent scheduling, we introduce a binary matrix $A \in \mathbb{R}^{n\times p}$ to depict the scheduling results. Specifically, $a_{ij} = 1$ when agent i is dispatched to relief site j, and $a_{ij} = 0$ otherwise. Concurrently, we establish a distance matrix $D \in \mathbb{R}^{n\times p}$, wherein d_{ij} represents the flying distance between agent i and relief site j. The fundamental aim of this research is to achieve the minimization of the average flying distance for agents to reach relief sites while maintaining the satisfaction of all relief site requirements.

Based on the description provided, the evaluation function for the task-scheduling approach presented in this paper is formulated to accurately assess its performance, which is expressed as:

$$\begin{aligned} &min \frac{\sum_{i=1}^{n} \sum_{j=1}^{p} a_{ij} d_{ij}}{n} \\ &st.\ m \sum_{i=1}^{n} a_{ij} \geq q_j \ j = 1, 2, ..., p. \end{aligned} \tag{4}$$

The dynamic model chosen for the nonlinear multi-agent system is outlined below:

$$\begin{cases} \dot{x}_i &= f_i(x) + x_{i+1} + d_i(t), \\ \dot{x}_n &= f_n(x) + u + d_n(t), \\ y &= x_1, \end{cases} \tag{5}$$

where u and y represent the input and output of the system, respectively, $x_i (i = 1, 2, ..., n)$ represents the system state variable, $d_i(t)$ represents the external disturbance bounded by an unknown positive constant d_i^*, $f_i(x) = \Phi_i^{*\mathrm{T}} \lambda(x) + \varepsilon_i$ denotes the unknown system nonlinear dynamic, and ε_i is the minimum approximation error.

3. Scheduling Policy Optimization Method Based on PPO

In rescue operations involving agents, the scheduling strategy of agents is crucial for enhancing rescue effectiveness. Especially under resource constraints, the challenge lies in how to allocate agents fairly and efficiently. To address this issue, this paper introduces the PPO algorithm and explores its application in optimizing agent-scheduling strategies. The PPO algorithm, with its distinctive advantages such as swift learning speed and robust convergence performance, has offered formidable support for tackling this intricate problem. Its capability to efficiently traverse the vast search space and converge to optimal solutions, even in high-dimensional problems, underscores its applicability and significance in addressing complex decision-making challenges within the context of agent rescue operations.

To address the agent-scheduling strategy optimization problem, we adopt a Markov Decision Process (MDP) framework. In detail, the specifics are as follows:

State: The state is framed as an $n \times p$ matrix, encoding the enclosure status between agents and targets. Specifically, if the i-th agent is scheduling the j-th target at time t, $(S_t)_{ij}$ takes a value of 1; otherwise, it is 0.

Action: The action space comprises a vector Ac of length p, where each element represents the allocation of agents to targets. Specifically, $Ac[i]$ indicates the identity of the agent assigned to surround the i-th target, thereby defining the scheduling strategy.

Reward: To optimize the multiple target assignment process using PPO, our primary goal is to maximize the efficiency of surrounding. This is achieved by defining a reward function as outlined in (4).

Return: The total return R_t is calculated as the discounted sum of all future rewards, starting from $t + 1$ and extending indefinitely. This is formally expressed as $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, where γ serves as the discount factor, set to 0.9 in this context, balancing the importance of immediate and future rewards.

The scheduling strategy optimization method based on PPO aims to iteratively seek the optimal scheduling strategy. The optimization of the agent-scheduling policy based on PPO is shown in Algorithm 1. It entails a systematic reinforcement learning process that initializes policy and value networks, iteratively collects data through agent–environment interactions, updates policy using a clipped surrogate loss function and importance sampling for stability and efficiency, concurrently refines the value function, and repeats this cycle to gradually improve scheduling strategies, harnessing PPO's strengths in stability, efficiency, and flexibility. The specific process is as follows:

Algorithm 1 The optimization of the agent-scheduling policy based on PPO.

Initialize:
Number of agents: n
Policy network $\pi(a|s;\theta)$ with parameters θ
Value network $V(s;\theta')$ with parameters θ'
Replay buffer D
Learning rates α_{actor}, α_{critic}
Discount factor γ
Clipping factor ϵ
Number of iterations T
Batch size B
for $t = 1$ to T **do**
    Clear micro-batch data B_t for the current iteration
    for each episode **do**
        Initialize environment state s (e.g., locations and demands of all sites)
        for each agent $u = 1$ to n **do**
            Select action a_u (allocation of rescue packages) based on $\pi(a|s;\theta)$
            Execute a_u (allocate packages to sites)
            Receive reward r_u based on satisfaction of demands
            Observe next state s'
            Store (s, a_u, r_u, s') in D
            Update $s \leftarrow s'$
        end for
    end for
    for each training step **do**
        Sample a batch B_t randomly from D
        for each $(s, a_u, r_u, s') \in B_t$ **do**
            Compute advantage function $A(s, a_u)$
            Compute clipped surrogate objective $L_t(\theta)$
            Update policy network parameters θ using gradient ascent: $\theta \leftarrow \theta + \alpha_{\text{actor}} \nabla_\theta L_t(\theta)$
            Compute value network target $V_{\text{target}} = r_u + \gamma V(s';\theta')$
            Update value network parameters θ using mean squared error loss: $\theta \leftarrow \theta - \alpha_{\text{critic}} \nabla_\theta (V(s;\theta) - V_{\text{target}})^2$
        end for
    end for
    Evaluate the current policy $\pi(a|s;\theta)$ on a validation set
    if performance does not improve significantly **then**
        Adjust hyperparameters or improve the network structure
    end if
end for
Output: The trained policy network $\pi(a|s;\theta)$ for agent rescue package allocation

Step 1. Initialization of Policy Network: In the initial phase, we establish a randomly initialized policy network, typically a neural network with parameters θ. This network takes the current state s as input and outputs two values: the probability distribution $\pi(a|s;\theta)$ for the action a (i.e., the scheduling decision) and the state value function $V(s;\theta)$. The objective of the policy network is to maximize the expected value of the cumulative rewards.

Step 2. Data Collection: In each iteration, we interact with the environment using the current policy network. Given the state s_t, the policy network selects an action a_t based on the probability distribution $\pi(a|s_t;\theta)$ and executes it. The environment then responds with the next state s_{t+1} and an immediate reward r_t. We collect these data points (s_t, a_t, r_t, s_{t+1}) to form trajectories, which are utilized for subsequent policy updates.

Step 3. Calculation of Advantage Function: The advantage function $A_\pi(s,a)$ represents the additional reward that can be obtained by taking action a in a given state s compared to the average reward obtained by following the current policy π. It can be as follows:

$$A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s) \tag{6}$$

where $Q_\pi(s,a)$ is the action-value function, representing the expected cumulative reward obtained by following policy π after taking action a in state s and $V_\pi(s)$ is the state-value function, representing the expected cumulative reward obtained by following policy π in state s.

In this paper, we estimate the advantage function using the Generalized Advantage Estimation (GAE) approach. GAE estimates the advantage function by introducing the φ-return method, which combines multi-step Temporal Difference (TD) errors. The TD error δ_t is defined as follows:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t), \tag{7}$$

where r_t is the immediate reward, $V(s_{t+1})$ and $V(s_t)$ are the value function estimates for the next and current states, respectively, and γ is the discount factor.

GAE calculates $A_t^{\mathrm{GAE}(\varphi)}$ as a weighted sum of multi-step TD errors, specifically

$$A_t^{\mathrm{GAE}(\varphi)} = \sum_{l=0}^{k-1} (\gamma\varphi)^l \delta_{t+l}, \tag{8}$$

where k is the number of steps considered and γ is the GAE's hyperparameter that balances bias and variance. When γ is close to 1, GAE favors multi-step returns, thus reducing variance; when γ is close to 0, GAE favors one-step returns, thus reducing bias.

Step 4. Update of Policy Network: Based on the PPO algorithm, we utilize the collected data and computed advantage function values to update the policy network. The PPO algorithm ensures stable updates by limiting the magnitude of changes between the old and new policies. Specifically, it employs a clipped surrogate objective function to update the parameters θ of the policy network, which can be expressed as follows:

$$L^{\mathrm{clip}}(\theta) = \mathrm{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{9}$$

where $r_t(\theta) = \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{old})}$ is the ratio of the probability of taking action a_t under the new policy to that under the old policy, $\hat{A}_t$ is an estimate of the advantage function $A_\pi(s_t,a_t)$, and ϵ is a hyperparameter that controls the degree of change between the old and new policies.

We iteratively repeat the process from Step 2 to Step 4 until a predefined number of iterations is reached or a convergence criterion is satisfied. During this iterative procedure, the policy network gradually converges to an optimal scheduling strategy, which maximizes the expected cumulative reward.

4. Predefined-Time Control Design and Stability Analysis

We define the error of the system as follows:

$$\begin{cases} \mu_1 = y - y_d, \\ \mu_i = x_i - \beta_{i-1}, \end{cases} \tag{10}$$

where μ_1 represents the tracking error, y_d stands for the desired command signal, μ_i refers to the virtual error variable, and β_{i-1} signifies the intermediate control variable.

The following Lyapunov function is to be chosen:

$$V_1 = \frac{1}{2}\mu_1^2 + \frac{1}{2}\tilde{\Phi}_1^2, \tag{11}$$

where $\tilde{\Phi}_1 = \Phi_1^* - \hat{\Phi}$, $\hat{\Phi}$ is the estimated value of parameter Φ_1^*.

From (5), (10), and (11), one can obtain the following:

$$\dot{V}_1 = \mu_1\left[\Phi_1^{*\mathrm{T}}\lambda_1(x) + \varepsilon_1 + \mu_2 + \beta_1 + d_1(t) - \dot{y}_d\right] - \tilde{\Phi}_1\dot{\hat{\Phi}}_1. \tag{12}$$

Additionally, in order to achieve the desired functionality, we can establish the intermediate control variable β_1 and a strategy for the adaptive update of $\hat{\Phi}_1$ as specified below:

$$\begin{aligned}\beta_1 = &-\frac{\varphi\pi}{\theta T_d}\mu_1^{1+\theta}\tanh\left(\frac{\varphi\pi}{\theta T_d}\frac{\mu_1^{2+\theta}}{\nu_1}\right) + \dot{y}_d \\ &-\frac{\overline{\varphi}c_0\pi}{\theta T_d}\mu_1^{1-\theta}\tanh\left(\frac{\overline{\varphi}c_0\pi}{\theta T_d}\frac{\mu_1^{2-\theta}}{\nu_1}\right) - \mu_1\chi_1\hat{\Phi}_1,\end{aligned} \tag{13}$$

$$\dot{\hat{\Phi}}_1 = -\gamma\frac{\varphi\pi}{\theta T_d}\hat{\Phi}_1^{1+\theta} - \overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\hat{\Phi}_1^{1-\theta} + \mu_1^2\chi_1, \quad \hat{\Phi}_1(0) > 0, \tag{14}$$

where θ is determined by dividing the positive even integer θ_1 (greater than the positive odd integer θ_2) by θ_2. The time constant T_d serves as a positive design parameter. The values of $c_0, \varphi, \bar{\varphi}, \gamma$, and $\bar{\gamma}$ are set to $\sqrt{2}+1, (2n)^{\frac{\theta}{2}}/2^{1+\frac{\theta}{2}}, 1/2^{1-\frac{\theta}{2}}, \gamma = 2+\theta, 2-\theta$.

By incorporating Equations (13) and (14) into Equation (12) and subsequently applying the corresponding inequality, we obtain

$$|w| - w\tanh(w/\nu) \le \tau\nu, \tag{15}$$

where ν is a positive, intentionally chosen parameter and τ is equal to 0.2785, the desired outcome can be achieved.

$$\begin{aligned}\dot{V}_1 \le &-\frac{\varphi\pi}{\theta T_d}\mu_1^{2+\theta}\tanh\left(\frac{\varphi\pi}{\theta T_d}\frac{\mu_1^{2+\theta}}{\nu_1}\right) + \gamma\frac{\varphi\pi}{\theta T_d}\tilde{\Phi}_1\hat{\Phi}_1^{1+\theta} + \mu_1\mu_2 \\ &-\frac{\overline{\varphi}c_0\pi}{\theta T_d}\mu_1^{2-\theta}\tanh\left(\frac{\overline{\varphi}c_0\pi}{\theta T_d}\frac{\mu_1^{2-\theta}}{\nu_1}\right) + \overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\tilde{\Phi}_1\hat{\Phi}_1^{1-\theta} + \frac{3}{4} \\ \le &-\frac{\varphi\pi}{\theta T_d}\left|\mu_1^{2+\theta}\right| + \gamma\frac{\varphi\pi}{\theta T_d}\tilde{\Phi}_1\hat{\Phi}_1^{1+\theta} + \mu_1\mu_2 \\ &-\frac{\overline{\varphi}c_0\pi}{\theta T_d}\left|\mu_1^{2-\theta}\right| + \overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\tilde{\Phi}_1\hat{\Phi}_1^{1-\theta} + \Theta_1,\end{aligned} \tag{16}$$

where $\Theta_1 = \frac{3}{4} + 2\tau\nu_1$.

In a similar manner to (11), we have selected the following Lyapunov function:

$$V_i = V_{i-1} + \frac{1}{2}\mu_i^2 + \frac{1}{2}\tilde{\Phi}_i^2, \tag{17}$$

Subsequently, we are able to derive

$$\begin{aligned}\dot{V}_i = \dot{V}_{i-1} + \mu_i\Bigg\{&\Phi_i^{*\mathrm{T}}\lambda_i(x) + \varepsilon_i + \mu_{i+1} + \beta_i + d_i(t) - H_{i-1} \\ &-\sum_{j=1}^{i-1}\frac{\partial\beta_{i-1}}{\partial x_j}\left[\Phi_j^{*\mathrm{T}}\lambda_j(x) + \varepsilon_j + d_j(t)\right]\Bigg\} - \tilde{\Phi}_i\dot{\hat{\Phi}}_i.\end{aligned} \tag{18}$$

Similarly, we define the intermediate control variables β_i and the adaptive update strategies $\hat{\Phi}_i$ as follow:

$$\begin{aligned}\beta_i = &-\frac{\varphi\pi}{\theta T_d}\mu_i^{1+\theta}\tanh\left(\frac{\varphi\pi}{\theta T_d}\frac{\mu_i^{2+\theta}}{\nu_i}\right) + H_{i-1} - \mu_i\chi_i\widehat{\Phi}_i \\ &-\frac{\overline{\varphi}c_0\pi}{\theta T_d}\mu_i^{1-\theta}\tanh\left(\frac{\overline{\varphi}c_0\pi}{\theta T_d}\frac{\mu_i^{2-\theta}}{\nu_i}\right) - \mu_{i-1},\end{aligned} \tag{19}$$

$$\dot{\widehat{\Phi}}_i = -\gamma\frac{\varphi\pi}{\theta T_d}\widehat{\Phi}_i^{1+\theta} - \overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\widehat{\Phi}_i^{1-\theta} + \mu_i^2\chi_i, \quad \widehat{\Phi}_i(0) > 0. \tag{20}$$

By incorporating Equations (19) and (20) into Equation (18) and utilizing the inequality stated in (15), one can obtain

$$\begin{aligned}\dot{V}_i \leq &-\sum_{j=1}^{i-1}\frac{\varphi\pi}{\theta T_d}\left|\mu_j^{2+\theta}\right| + \sum_{j=1}^{i}\gamma\frac{\varphi\pi}{\theta T_d}\widetilde{\Phi}_j\widehat{\Phi}_j^{1+\theta} \\ &-\sum_{j=1}^{i-1}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\left|\mu_j^{2-\theta}\right| + \sum_{j=1}^{i}\overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\widetilde{\Phi}_j\widehat{\Phi}_j^{1-\theta} \\ &-\frac{\varphi\pi}{\theta T_d}\mu_i^{2+\theta}\tanh\left(\frac{\varphi\pi}{\theta T_d}\frac{\mu_i^{2+\theta}}{\nu_i}\right) \\ &-\frac{\overline{\varphi}c_0\pi}{\theta T_d}\mu_i^{2-\theta}\tanh\left(\frac{\overline{\varphi}c_0\pi}{\theta T_d}\frac{\mu_i^{2-\theta}}{\nu_i}\right) + \mu_i\mu_{i+1} + \Theta_{i-1} + \frac{3i}{4} \\ \leq &-\sum_{j=1}^{i}\frac{\varphi\pi}{\theta T_d}\left|z_j^{2+\theta}\right| + \sum_{j=1}^{i}\gamma\frac{\varphi\pi}{\theta T_d}\widetilde{\Phi}_j\widehat{\Phi}_j^{1+\theta} + \mu_i\mu_{i+1} \\ &-\sum_{j=1}^{i}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\left|z_j^{2-\theta}\right| + \sum_{j=1}^{i}\overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\widetilde{\Phi}_j\widehat{\Phi}_j^{1-\theta} + \Theta_i,\end{aligned} \tag{21}$$

where $\Theta_i = \Theta_{i-1} + \frac{3i}{4} + 2\tau\nu_i$.

Consistent with our approach, we adopt the Lyapunov function specified below:

$$V_n = V_{n-1} + \frac{1}{2}\mu_n^2 + \frac{1}{2}\widetilde{\Phi}_n^2, \tag{22}$$

then we can obtain

$$\begin{aligned}\dot{V}_n = &\dot{V}_{n-1} + \mu_n\Big\{\Phi_n^{*\mathrm{T}}\lambda_n(x) + \varepsilon_n + u + d_n(t) - H_{n-1} \\ &-\sum_{j=1}^{n-1}\frac{\partial\beta_{n-1}}{\partial x_j}\left[\Phi_j^{*\mathrm{T}}\lambda_j(x) + \varepsilon_j(x) + d_j(t)\right]\Big\} - \widetilde{\Phi}_n\dot{\widehat{\Phi}}_n.\end{aligned} \tag{23}$$

Similarly, we define the predefined-time controller u and the adaptive update strategies $\hat{\Phi}_n$ as follows:

$$\begin{aligned}u = &-\frac{\varphi\pi}{\theta T_d}\mu_n^{1+\theta}\tanh\left(\frac{\varphi\pi}{\theta T_d}\frac{\mu_n^{2+\theta}}{\nu_n}\right) + H_{n-1} - \mu_n\chi_n\widehat{\Phi}_n \\ &-\frac{\overline{\varphi}c_0\pi}{\theta T_d}\mu_n^{1-\theta}\tanh\left(\frac{\overline{\varphi}c_0\pi}{\theta T_d}\frac{\mu_n^{2-\theta}}{\nu_n}\right) - z_{n-1},\end{aligned} \tag{24}$$

$$\dot{\widehat{\Phi}}_n = -\gamma\frac{\varphi\pi}{\theta T_d}\widehat{\Phi}_n^{1+\theta} - \overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\widehat{\Phi}_n^{1-\theta} + \mu_n^2\chi_n, \quad \widehat{\Phi}_n(0) > 0. \tag{25}$$

Substituting (24) and (25) into (23) and applying the inequality in (15), one can obtain

$$\begin{aligned}\dot{V}_n \leq & -\sum_{j=1}^{n-1}\frac{\varphi\pi}{\theta T_d}\left|\mu_j^{2+\theta}\right| + \sum_{j=1}^{n}\gamma\frac{\varphi\pi}{\theta T_d}\widetilde{\Phi}_j\widehat{\Phi}_j^{1+\theta} \\ & -\sum_{j=1}^{n-1}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\left|\mu_j^{2-\theta}\right| + \sum_{j=1}^{n}\overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\widetilde{\Phi}_j\widehat{\Phi}_j^{1-\theta} \\ & -\frac{\varphi\pi}{\theta T_d}\mu_n^{2+\theta}\tanh\left(\frac{\varphi\pi}{\theta T_d}\frac{\mu_n^{2+\theta}}{\nu_n}\right) \\ & -\frac{\overline{\varphi}c_0\pi}{\theta T_d}\mu_n^{2-\theta}\tanh\left(\frac{\overline{\varphi}c_0\pi}{\theta T_d}\frac{\mu_n^{2-\theta}}{\nu_n}\right) + \Theta_{n-1} + \frac{3n}{4} \\ \leq & -\sum_{j=1}^{n}\frac{\varphi\pi}{\theta T_d}\left|\mu_j^{2+\theta}\right| + \sum_{j=1}^{n}\gamma\frac{\varphi\pi}{\theta T_d}\widetilde{\Phi}_j\widehat{\Phi}_j^{1+\theta} \\ & -\sum_{j=1}^{n}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\left|\mu_j^{2-\theta}\right| + \sum_{j=1}^{n}\overline{\gamma}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\widetilde{\Phi}_j\widehat{\Phi}_j^{1-\theta} + \Theta_n,\end{aligned} \tag{26}$$

where $\Theta_n = \Theta_{n-1} + \frac{3n}{4} + 2\tau\nu_n$.

The following inequalities can be derived from [31,32]:

$$\gamma\widetilde{\Phi}_j\widehat{\Phi}_j^{1+\theta} \leq \Phi_j^{*(2+\theta)} - \widehat{\Phi}_j^{2+\theta} \leq 2\Phi_j^{*(2+\theta)} - \widetilde{\Phi}_j^{2+\theta}, \tag{27}$$

$$\overline{\gamma}\widetilde{\Phi}_j\widehat{\Phi}_j^{1-\theta} \leq \Phi_j^{*(2-\theta)} - \widehat{\Phi}_j^{2-\theta} \leq 2\Phi_j^{*(2-\theta)} - \widetilde{\Phi}_j^{2-\theta}. \tag{28}$$

Substituting (27) and (28) into (26), one has

$$\begin{aligned}\dot{V}_n \leq & -\sum_{j=1}^{n}\frac{\varphi\pi}{\theta T_d}\left|\mu_j^{2+\theta}\right| - \sum_{j=1}^{n}\frac{\varphi\pi}{\theta T_d}\widetilde{\Phi}_j^{2+\theta} \\ & -\sum_{j=1}^{n}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\left|\mu_j^{2-\theta}\right| - \sum_{j=1}^{n}\frac{\overline{\varphi}c_0\pi}{\theta T_d}\widetilde{\Phi}_j^{2-\theta} + \Theta,\end{aligned} \tag{29}$$

where $\Theta = \Theta_n + \sum_{j=1}^{n}\frac{\pi(2n)^{\frac{\theta}{2}}}{2^{\frac{\theta}{2}}\theta T_d}\Phi_j^{*(2+\theta)} + \sum_{j=1}^{n}\frac{c_0\pi}{2^{-\frac{\theta}{2}}\theta T_d}\Phi_j^{*(2-\theta)}$.

According to [34**?**], for any $x_i \in R$, there exist constants p and q satisfying $0 < p \leq 1$ and $1 < q \leq \infty$, for which the following inequalities are valid:

$$\left(\sum_{i=1}^{n}|x_i|\right)^p \leq \sum_{i=1}^{n}|x_i|^p, \quad \left(\sum_{i=1}^{n}|x_i|\right)^q \leq n^{q-1}\sum_{i=1}^{n}|x_i|^q. \tag{30}$$

Then, one can obtain

$$\begin{aligned}\dot{V}_n \leq & -\frac{\pi}{\theta T_d}\left[\sum_{j=1}^{n}\left(\frac{1}{2}\mu_j^2 + \frac{1}{2}\widetilde{\Phi}_j^2\right)\right]^{1+\frac{\theta}{2}} \\ & -\frac{c_0\pi}{\theta T_d}\left[\sum_{j=1}^{n}\left(\frac{1}{2}\mu_j^2 + \frac{1}{2}\widetilde{\Phi}_j^2\right)\right]^{1-\frac{\theta}{2}} + \Theta \\ = & -\frac{\pi}{\theta T_d}V_n^{1+\frac{\theta}{2}} - \frac{c_0\pi}{\theta T_d}V_n^{1-\frac{\theta}{2}} + \Theta.\end{aligned} \tag{31}$$

5. Simulation Setup And Result Analysis

5.1. Simulation Setup

To validate the effectiveness of the theoretical derivation and the proposed agent-scheduling optimization strategy, this section designs and executes a series of simulation experiments using the Python programming language. To comprehensively consider the potential impact of the number of agents and the scale of rescue areas on algorithm performance, we specifically set up two experimental scenarios with different characteristics to demonstrate the versatility and adaptability of the algorithm under various conditions. In both scenarios, each agent is set to carry one rescue package as the task load. Specifically, scenario one simulates a multi-agent system consisting of ten agents and three rescue areas, with the initial positions of the agents and the locations of the rescue targets detailed in Tables 1 and 2, respectively. Scenario two further expands the system complexity by introducing fourteen agents and four rescue areas, and, similarly, the initial deployment of agents and the locations of task targets are set according to Tables 3 and 4.

Table 1. The initial position of the agent in Scenario 1.

Agent	Lat	Lon	Agent	Lat	Lon
1	0.0426	1.0557	6	4.5103	7.9949
2	2.8584	0.2695	7	1.4984	4.0147
3	4.7161	0.6012	8	0.5418	4.5941
4	7.7187	7.4370	9	1.7561	9.4918
5	5.9442	8.8786	10	8.4732	8.7486

Table 2. The information of rescue areas in Scenario 1.

Rescue Area	Lat	Lon	Demand
1	6.4831	2.1476	3
2	9.4930	0.1211	3
3	1.8088	1.8772	4

Table 3. The initial position of the agent in Scenario 2.

Agent	Lat	Lon	Agent	Lat	Lon
1	6.1470	3.8101	8	7.1160	2.0503
2	6.3711	4.7446	9	3.0778	9.8086
3	4.4253	0.9577	10	0.1026	4.6604
4	6.1416	0.5733	11	4.6038	8.5465
5	5.6571	5.3323	12	4.5247	6.3166
6	3.9005	9.0885	13	4.7600	2.2003
7	5.3337	7.0734	14	7.1359	6.1904

Table 4. The information of rescue areas in Scenario 2.

Rescue Area	Lat	Lon	Demand
1	2.1660	2.5708	3
2	0.4580	1.7551	3
3	6.1767	8.2907	4
4	5.2463	2.7081	4

This paper simulation platform is *PyTorch*, and the specific parameter configuration of PPO is given in Table 5. And the structure of the PPO networks is given in Table 6 To demonstrate the outstanding performance of the method proposed in this study, we conducted an in-depth comparison between the PPO algorithm and the classic DQN, a traditional optimization technique. This move aims to validate the significant advantages of the PPO algorithm in terms of optimization efficiency and effectiveness through direct performance evaluation.

Table 5. Hyperparameter configuration.

Notation	Meaning
Time steps	20,000
Number of agents	10/14
Number of rescue areas	3/4
Actor learning rate	0.01
Critic learning rate	0.001
Discount	0.9

Table 6. Structure of the PPO networks.

Policy Network	Value Network
Input(c,state_dim)	Input(c,state_dim)
Linear(state_dim,hidden_dim)	Linear(state_dim,hidden_dim)
relu	softmax
Linear(hidden_dim,1)	Linear(hidden_dim,1)

For the singularity-free predefined-time fuzzy adaptive tracking control strategy based on nonlinear systems, in (13) and (14) and (24) and (25), we set $T_d = 5$, $\theta = \frac{\theta_1}{\theta_2} = \frac{6}{75}$ and $\nu_1 = \nu_2 = 0.001$.

$$\lambda_c(x) = \frac{\prod_{i=1}^{b} \varpi_{B_i^c}(x_i)}{\sum_{c=1}^{\iota}\left[\prod_{i=1}^{b} \varpi_{B_i^c}(x_i)\right]}. \tag{32}$$

The fuzzy membership functions are selected as

$$\varpi_{B_1^c} = \mathrm{e}^{-\frac{(x_1-3+c)^2}{16}}, \quad \varpi_{B_2^c} = \mathrm{e}^{-\frac{(x_2-6+2c)^2}{4}}, \tag{33}$$

where the number of fuzzy rules is $\iota = 5$ and $b = 1, 2$.

5.2. Result Analysis

Figures 1 and 2 comprehensively exhibit the comparative results of agent scheduling under two distinct scenarios, aiming to delve into and compare the performance of two scheduling strategies. Specifically, Figure 1a visually presents the detailed layout of task allocation among agents under the random scheduling strategy in a scenario with ten agents and three rescue areas, where the average distance between agents is 6.61. In contrast, Figure 1b reveals the optimized state of task allocation and a more compact layout among agents in the same scenario after adopting the PPO-optimized scheduling strategy, with the average distance reduced to 5.40. Furthermore, Figure 2a,b showcase the contrast between the effects of random scheduling and PPO-optimized scheduling in a complex scenario expanded to fourteen agents and four rescue areas, with the latter significantly decreasing the average distance between agents from 4.88 to 3.72. This series of comparative analyses not only highlights the superiority of the PPO algorithm in shortening the average distance from agents to rescue areas, but also validates the algorithm's wide applicability and efficiency in solving problems through the setting of two different scenarios.

This figure, compared to the results under the random scheduling strategy, demonstrates a remarkably optimized effect. This finding not only validates the effectiveness of the PPO strategy in scheduling optimization, but also provides a strong basis for our future research and applications. By comparing and analyzing the performance of these two strategies, we can gain a clearer understanding of the differences in resource allocation and efficiency optimization between different strategies, providing valuable references for future research and applications.

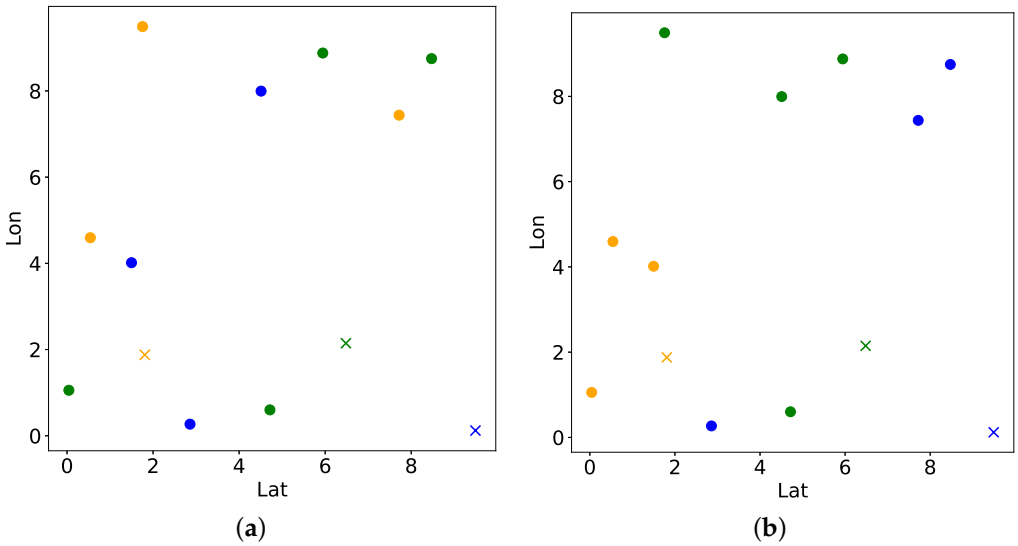


Figure 1. Agent scheduling results for Scenario 1. × represents the rescue area, • represents the agent. (**a**) The results of the random scheduling of agents. The average distance is 6.61. (**b**) The scheduling results after optimization using the PPO algorithm. The average distance is 5.40.

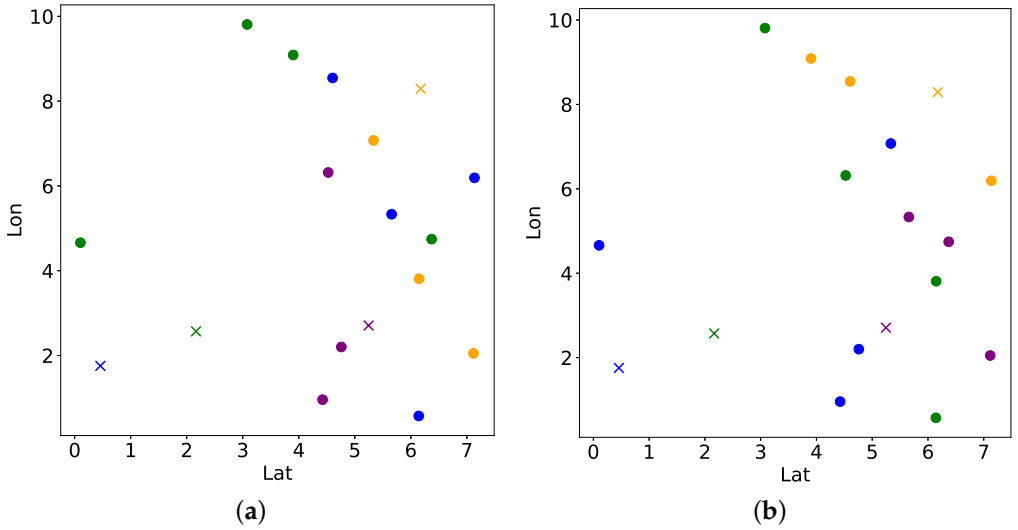


Figure 2. Agent scheduling results for Scenario 2. × represents the rescue area, • represents the agent. (**a**) The results of the random scheduling of agents. The average distance is 4.88. (**b**) The scheduling results after optimization using the PPO algorithm. The average distance is 3.72.

This paper comparatively analyzes the performance of the PPO algorithm against the DQN algorithm, aiming to demonstrate the superiority of the proposed approach. Figure 3 vividly illustrates the distinct differences in average return values and respective trend characteristics between the two. Specifically, the figure encapsulates the learning progression of both algorithms under Scenario 2, as devised in this study, providing quanti-

tative evidence for assessing their efficiency and stability. Observing the DQN algorithm, representative of traditional methods, its average return exhibits a gradual decline from a peak of 10.5 during the initial 15,000 iterations, subsequently stabilizing over the next 5000 iterations to converge at approximately 7.9. This phenomenon underscores DQN's potential for overfitting in early learning phases and its relatively slower convergence rate, limiting performance enhancements in complex environments which require rapid adaptation and optimization. In contrast, the PPO algorithm demonstrates remarkable convergence efficiency and optimization prowess. As depicted in Figure 3, PPO's average return swiftly peaks at 9.5 before steadily declining and converging to around 6.2 after approximately 12,500 iterations, advancing 2500 iterations ahead of DQN. This outcome underscores PPO's efficiency in strategy optimization and underscores its unique advantage in maintaining training stability through constrained policy update steps. PPO's ability to swiftly identify and stabilize on a relatively superior strategy underscores its capacity to offer more efficient results.

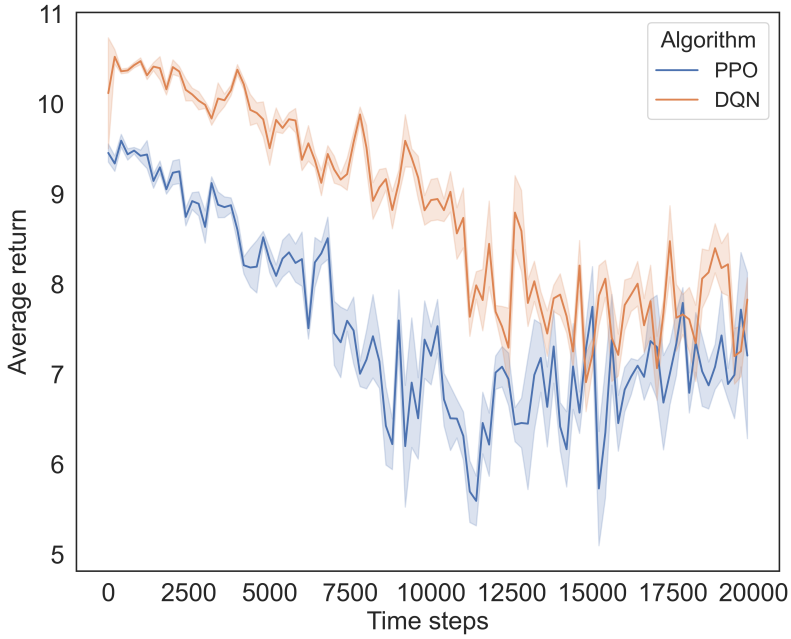


Figure 3. Comparison of the average return values between the PPO algorithm (in this paper) and the DQN algorithm (traditional).

The simulation results of the non-singular predefined-time fuzzy adaptive tracking control strategy based on nonlinear systems are presented in detail in Figures 4–7. Figure 4 presents a comparison of the trajectories of the system output signal y and the desired command signal y_d, demonstrating a high degree of overlap between the two signals, which visually showcases the dynamic response of the system. Figure 5 shows the trajectory of the tracking error, indicating that before the preset time $T_d = 5$, the tracking error μ_1 can stably converge to a very small range near zero. This result fully validates the effectiveness and accuracy of the control algorithm. As evident from Figures 4 and 5, the proposed preset-time control algorithm in this study exhibits excellent tracking control performance. Figure 6 presents the trajectory of the control system's input signal u, while Figure 7 depicts the evolution trajectory of the adaptive parameter Φ. This demonstrates that the proposed preset-time control method in this paper enables the agent to reach its destination within the desired time frame for rescue.

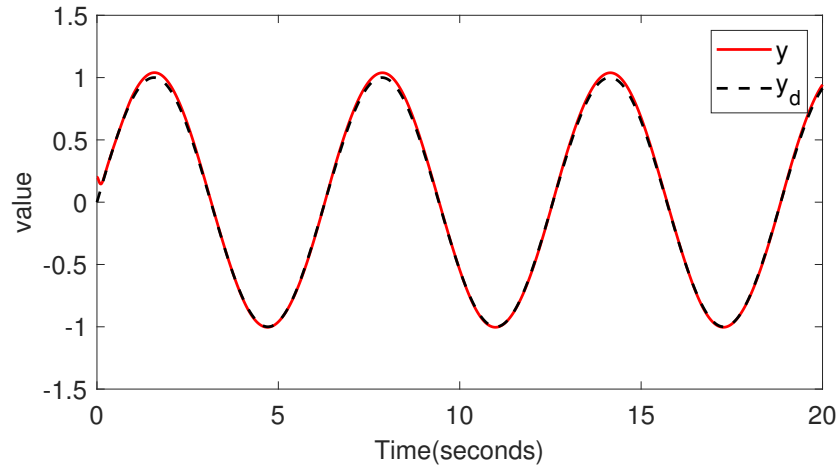


Figure 4. The trajectories of y and y_d.

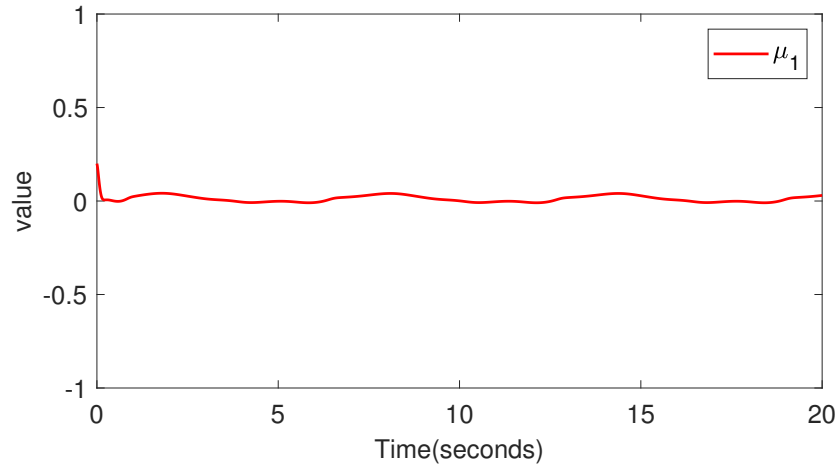


Figure 5. The trajectory of tracking error μ_1.

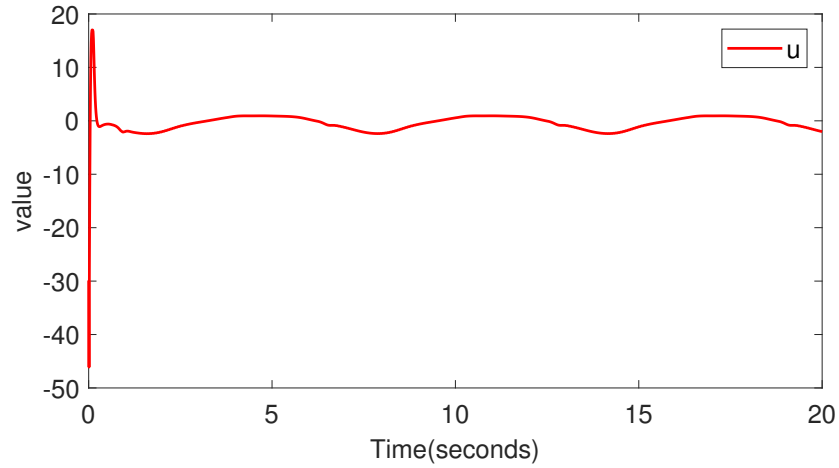


Figure 6. The trajectory of control input u.

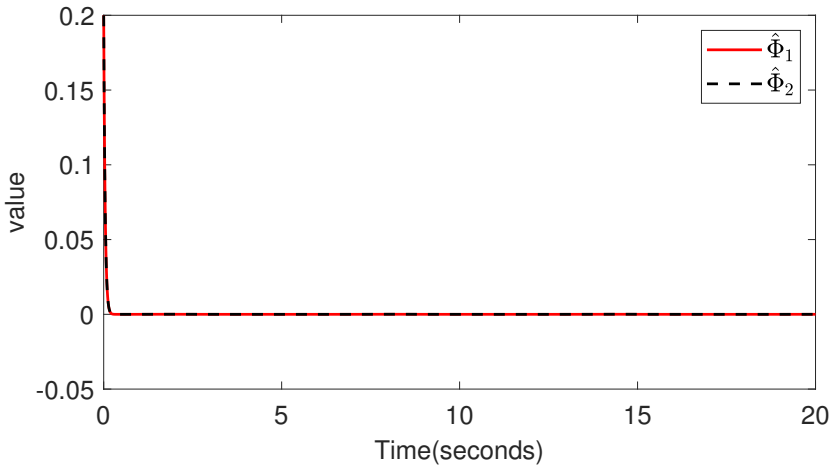


Figure 7. The trajectories of adaptive parameters $\hat{\Phi}_i (i = 1, 2)$.

6. Conclusions

In this paper, we introduce innovative scheduling and control strategies for the rescue missions of intelligent agents. We construct a function related to the average distance of the intelligent agents and design an optimization method for the scheduling strategy based on the PPO algorithm, achieving efficient response to rescue missions. Furthermore, we propose a singularity-free predefined-time fuzzy adaptive tracking control strategy based on nonlinear systems, ensuring that the intelligent agents can accurately reach designated locations within the predetermined time. Finally, two independent experimental scenarios are designed to comprehensively evaluate the universality and efficiency of the proposed method. Specifically, to highlight the superiority of the PPO algorithm in optimizing scheduling strategies, the PPO algorithm is directly compared with the classical DQN algorithm. The experimental results show that, when compared with the scheduling strategy optimized by the DQN algorithm, the strategy generated by the PPO algorithm significantly shortens the average travel distance for the agent to complete the task by 1.7 units, while achieving a 27% increase in efficiency, effectively demonstrating the superior performance of the PPO algorithm in the field of scheduling optimization. These innovative approaches not only improve the rescue efficiency, but also enhance the reliability of the intelligent agents in complex environments.

Author Contributions: Quantitative Analysis, Modeling: D.Q.; Method Validation and Simulation Scenario Design: Y.Z.; Method Design, Paper Editing, and Submission: L.L.; Simulation Processing and Analysis: Z.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (NSFC), Grant/Award Number: 72071209 and the Youth Talent Lifting Project of the China Association for Science and Technology No. 2021-JCJQ-QT-018.

Data Availability Statement: Data will be made available on request.

Acknowledgments: Thanks to the editors and reviewers for their help in improving the quality of the paper.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Cheng, J.; Gao, Y.; Tian, Y.; Liu, H. GA-LNS Optimization for Helicopter Rescue Dispatch. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3898–3912. [CrossRef]
2. Wu, J.; Song, C.; Ma, J.; Wu, J.; Han, G. Reinforcement Learning and Particle Swarm Optimization Supporting Real-Time Rescue Assignments for Multiple Autonomous Underwater Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6807–6820. [CrossRef]
3. Akgun, S.A.; Ghafurian, M.; Crowley, M.; Dautenhahn, K. Using Affect as a Communication Modality to Improve Human-Robot Communication in Robot-Assisted Search and Rescue Scenarios. *IEEE Trans. Affect. Comput.* **2023**, *14*, 3013–3030. [CrossRef]

4. Wang, Y.; Chen, X.; Wang, L. Deep Reinforcement Learning-Based Rescue Resource Distribution Scheduling of Storm Surge Inundation Emergency Logistics. *IEEE Trans. Ind. Inform.* **2023**, *19*, 10004–10013. [CrossRef]
5. Wu, J.; Sun, Y.; Li, D.; Shi, J.; Li, X.; Gao, L.; Yu, L.; Han, G.; Wu, J. An Adaptive Conversion Speed Q-Learning Algorithm for Search and Rescue UAV Path Planning in Unknown Environments. *IEEE Trans. Veh. Technol.* **2023**, *72*, 15391–15404. [CrossRef]
6. Huo, M.; Duan, H.; Zeng, Z. Cluster Space Control Method of Manned-Unmanned Aerial Team for Target Search Task. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *70*, 2545–2549. [CrossRef]
7. Ribeiro, R.G.; Cota, L.P.; Euzébio, T.A.M.; Ramírez, J.A.; Guimarães, F.G. Unmanned-Aerial-Vehicle Routing Problem with Mobile Charging Stations for Assisting Search and Rescue Missions in Postdisaster Scenarios. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 6682–6696. [CrossRef]
8. Zhang, J.; Jia, Y.; Zhu, D.; Hu, W.; Tang, Z. Study on the Situational Awareness System of Mine Fire Rescue Using Faster Ross Girshick-Convolutional Neural Network. *IEEE Intell. Syst.* **2020**, *35*, 54–61. [CrossRef]
9. Dell'Agnola, F.; Jao, P.K.; Arza, A.; Chavarriaga, R.; Millán, J.d.R.; Floreano, D.; Atienza, D. Machine-Learning Based Monitoring of Cognitive Workload in Rescue Missions with Drones. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 4751–4762. [CrossRef]
10. Men, J.; Jiang, P.; Zheng, S.; Kong, Y.; Zhao, Y.; Sheng, G.; Su, N.; Zheng, S. A Multi-Objective Emergency Rescue Facilities Location Model for Catastrophic Interlocking Chemical Accidents in Chemical Parks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4749–4761. [CrossRef]
11. Rani, S.; Babbar, H.; Kaur, P.; Alshehri, M.D.; Shah, S.H. An Optimized Approach of Dynamic Target Nodes in Wireless Sensor Network Using Bio Inspired Algorithms for Maritime Rescue. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 2548–2555. [CrossRef]
12. Wang, L.; Zhao, X.; Wu, P. Resource-Constrained Emergency Scheduling for Forest Fires via Artificial Bee Colony and Variable Neighborhood Search Combined Algorithm. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 5791–5806. [CrossRef]
13. Yang, F.; Wu, N.; Qiao, Y.; Zhou, M.; Su, R.; Qu, T. Modeling and Optimal Cyclic Scheduling of Time-Constrained Single-Robot-Arm Cluster Tools via Petri Nets and Linear Programming. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 871–883. [CrossRef]
14. Sun, H.; Wang, S.; Zhou, F.; Yin, L.; Liu, M. Dynamic Deployment and Scheduling Strategy for Dual-Service Pooling-Based Hierarchical Cloud Service System in Intelligent Buildings. *IEEE Trans. Cloud Comput.* **2023**, *11*, 139–155. [CrossRef]
15. Wei, F.; Wan, Z.; He, H.; Lin, X.; Li, Y. A Novel Scheduling Strategy for Controllable Loads with Power-Efficiency Characteristics. *IEEE Trans. Smart Grid* **2020**, *11*, 2151–2161. [CrossRef]
16. Xie, J.; Huang, S.; Wei, D.; Zhang, Z. Scheduling of Multisensor for UAV Cluster Based on Harris Hawks Optimization with an Adaptive Golden Sine Search Mechanism. *IEEE Sens. J.* **2022**, *22*, 9621–9635. [CrossRef]
17. Du, W.; Zhong, W.; Tang, Y.; Du, W.; Jin, Y. High-Dimensional Robust Multi-Objective Optimization for Order Scheduling: A Decision Variable Classification Approach. *IEEE Trans. Ind. Inform.* **2019**, *15*, 293–304. [CrossRef]
18. Yue, H.; Zhang, Q.; Zeng, X.; Huang, W.; Zhang, L.; Wang, J. Optimal Scheduling Strategy of Electric Vehicle Cluster Based on Index Evaluation System. *IEEE Trans. Ind. Appl.* **2023**, *59*, 1212–1221. [CrossRef]
19. Du, Y.; Wang, T.; Xin, B.; Wang, L.; Chen, Y.; Xing, L. A Data-Driven Parallel Scheduling Approach for Multiple Agile Earth Observation Satellites. *IEEE Trans. Evol. Comput.* **2020**, *24*, 679–693. [CrossRef]
20. Min, J.; Kim, W.; Paek, J.; Govindan, R. Effective Routing and Scheduling Strategies for Fault-Tolerant Time-Sensitive Networking. *IEEE Internet Things J.* **2024**, *11*, 11008–11020. [CrossRef]
21. Zhang, W.; Shi, J.; Zhang, S.; Chen, M. Scenario-Based Robust Remanufacturing Scheduling Problem Using Improved Biogeography-Based Optimization Algorithm. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 3414–3427. [CrossRef]
22. Kumar, S.; Soni, S.K.; Kumar Pal, A.; Kamal, S.; Xiong, X. Nonlinear Polytopic Systems with Predefined Time Convergence. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 632–636. [CrossRef]
23. Becerra, H.M.; Vázquez, C.R.; Arechavaleta, G.; Delfin, J. Predefined-Time Convergence Control for High-Order Integrator Systems Using Time Base Generators. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 1866–1873. [CrossRef]
24. Wang, J.; Wong, W.C.; Luo, X.; Li, X.; Guan, X. Cooperative Platoon Control for Uncertain Networked Aerial Vehicles with Predefined-Time Convergence. *IEEE Internet Things J.* **2022**, *9*, 5982–5991. [CrossRef]
25. Sánchez Torres, J.D.; Gomez-Gutierrez, D.; López, E.; Loukianov, A. A class of predefined-time stable dynamical systems. *IMA J. Math. Control Inf.* **2017**, *35*, i1–i29. [CrossRef]
26. Jiménez-Rodríguez, E.; Muñoz-Vázquez, A.J.; Sánchez-Torres, J.D.; Loukianov, A.G. A Note on Predefined-Time Stability. *IFAC-PapersOnLine* **2018**, *51*, 520–525. [CrossRef]
27. Sánchez-Torres, J.D.; Muñoz-Vázquez, A.J.; Defoort, M.; Jiménez-Rodríguez, E.; Loukianov, A.G. A class of predefined-time controllers for uncertain second-order systems. *Eur. J. Control* **2020**, *53*, 52–58. [CrossRef]
28. Ni, J.; Liu, L.; Tang, Y.; Liu, C. Predefined-Time Consensus Tracking of Second-Order Multiagent Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 2550–2560. [CrossRef]
29. Cui, G.; Xu, H.; Xu, S.; Gu, J. Predefined-Time Adaptive Fuzzy Bipartite Consensus Control for Multiquadrotors Under Malicious Attacks. *IEEE Trans. Fuzzy Syst.* **2024**, *32*, 2187–2197. [CrossRef]
30. Xu, H.; Yu, D.; Sui, S.; Chen, C.L.P. An Event-Triggered Predefined Time Decentralized Output Feedback Fuzzy Adaptive Control Method for Interconnected Systems. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 631–644. [CrossRef]
31. Yang, H.; Ye, D. Adaptive fixed-time bipartite tracking consensus control for unknown nonlinear multi-agent systems: An information classification mechanism. *Inf. Sci.* **2018**, *459*, 238–254. [CrossRef]

32. Cui, G.; Xu, H.; Yu, J.; Lam, H.K. Event-Triggered Distributed Fixed-Time Adaptive Attitude Control with Prescribed Performance for Multiple QUAVs. *IEEE Trans. Autom. Sci. Eng.* **2023**, 1–11. [CrossRef]
33. Zuo, Z.; Tie, L. A new class of finite-time nonlinear consensus protocols for multi-agent systems. *Int. J. Control* **2014**, *87*, 363–370. [CrossRef]
34. Cui, G.; Xu, H.; Chen, X.; Yu, J. Fixed-Time Distributed Adaptive Formation Control for Multiple QUAVs with Full-State Constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 4192–4206. [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.