



Article A Drone Scheduling Method for Emergency Power Material Transportation Based on Deep Reinforcement Learning Optimized PSO Algorithm

Wenjiao Zai⁺, Junjie Wang^{*,†} and Guohui Li

College of Engineering, Sichuan Normal University, Chengdu 610101, China; zaiwenjiao@sicnu.edu.cn (W.Z.); guohuili@sicnu.edu.cn (G.L.)

* Correspondence: 20212193014@stu.sicnu.edu.cn; Tel.: +86-158-8457-8454

⁺ These authors contributed equally to this work.

Abstract: Stable material transportation is essential for quickly restoring the power system following a disaster. Drone-based material transportation can bypass ground transportation's limitations and reduce transit times. However, the current drone flight trajectory distribution optimization model cannot meet the need for mountainous emergency relief material distribution following a disaster. A power emergency material distribution model with priority conditions is proposed in this paper, along with a two-layer dynamic task-solving framework that takes task dynamics into account. This research proposes an algorithm (TD3PSO) that combines the particle swarm algorithm (PSO) updating technique with the double-delay depth deterministic policy gradient algorithm (TD3) algorithm's capacity to dynamically parameterize. The final task allocation experiment demonstrates that the modified TD3PSO significantly outperforms the conventional algorithm on the Solomon data set, with an improvement of 26.3% on average over the RLPSO algorithm and a 11.0% reduction in the volatility of the solving impact. When solving under realistic circumstances, the solution effect increases by 1.6% to 13.4%, and the redistribution experiment confirms the framework's efficacy. As a result, the algorithm and architecture suggested in this paper may successfully address the issue of scheduling drones for power emergencies while enhancing transportation efficiency.

Keywords: power emergency material distribution model; drone flight trajectory; task allocation; PSO; the double-delay depth deterministic policy gradient algorithm

1. Introduction

Creating rational emergency distribution plans can significantly speed up the distribution system's recovery [1,2]. Traditional emergency distribution relies heavily on the transportation network, which is susceptible to significant damage due to the mountainous regions of western China's frequent natural catastrophes and complicated topographical conditions [3]. Drone airdrop material techniques can successfully address these issues.

How to perform fast drone flight trajectory assignment is one of the keys to solving the problem. The Vehicle Routing Problem of Drones (VRPD), a variation of the Vehicle Path Problem (VRP), has been researched by researchers. In the classic VRPD problem, Different needs and the actual situation can determine the objective function, such as the shortest transport delivery time of the material [4–6] and the time window constraints [6]. However, many of the current studies are poorly targeted, abstracted only to conventional material transport problems, ignoring the distinctions between various transport problems and the dynamics of the problem, which would be more pertinent if the target point's characteristics were taken into account in the text.

Finding the best solution to the VRPD problem is a significant challenge. Heuristic algorithms are more efficient at solving the VRPD problem in emergency settings [7], such as the Particle Swarm Algorithm (PSO) [8], the Genetic Algorithm (GA) [9], and the



Citation: Zai, W.; Wang, J.; Li, G. A Drone Scheduling Method for Emergency Power Material Transportation Based on Deep Reinforcement Learning Optimized PSO Algorithm. *Sustainability* **2023**, *15*, 13127. https://doi.org/ 10.3390/su151713127

Academic Editors: Marie-Ange Manier and Hervé Manier

Received: 30 July 2023 Revised: 17 August 2023 Accepted: 29 August 2023 Published: 31 August 2023



Copyright: © 2023 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/). Dynamical Artificial Bee Colony (DABC) [10] algorithm. Among these algorithms, the PSO has the benefits of simple computing and quick convergence and is appropriate for application in rapid-demand emergency rescue work allocation [11]. The fast decline in population variety as the number of iterations variegation rises makes PSO extremely vulnerable to local optima and highly reliant on parameter settings.

This study will provide a UAV power transportation distribution scheme to provide a feasible solution for restoring the power system in mountainous areas after a disaster and improving distribution efficiency. The main contributions of this study are as follows:

- 1. This paper proposes a cooperative scheduling model for emergency material UAV swarms applied to the power system, which effectively provides a reference for emergency rescue UAV transportation in the power system.
- 2. In contrast to the conventional linear gradient update PSO, roulette-PSO(PSO-), Multi-Group Particle Swarm Optimization with Random Redistribution (MGRR)-PSO [12], and Reinforcement Learning (RL)-PSO [13], a TD3PSO algorithm for solving VRPD is proposed. This approach effectively increases the stability of the algorithm and the solving impact.
- Classification ideas are used to classify the redistribution tasks, and TD3PSO combined with the node-centered method is applied to solve the problem. This framework and solving method solve the dynamic allocation problem of power emergency supplies UAV and improve the operation efficiency.

The main idea of this research is expressed in Figure 1. Based on this, the remainder of the essay is structured as follows: The associated literature is reviewed in Section 2. The task model and redistribution framework are created in Section 3 using the problem description as a starting point. The TD3PSO algorithm's design is presented in Section 4. The algorithm's validation and a case study are covered in Section 5. There is a discussion in Section 6, and future research is combined and mentioned in Section 7.



Figure 1. TD3PSO-based realization scheme of uncrewed aerial vehicle (UAV) emergency material distribution in mountainous areas. (**a**): The data collection part of the paper; (**b**): Data processing section, which processes the collected data and creates 2D maps; (**c**): This part is the objective function, which consists of a mathematical model and power priority nodes; (**d**): The td3pso algorithm solution framework; (**e**): The task reassignment framework, and the task solution process e-process is not necessary; (**f**): The output of the algorithm (UAV direction trajectory, Gantt chart, table of task assignment results).

2. Related Work

This section summarizes the ongoing research on the issue at hand, including the distribution of power emergency materials, the VRPD, and the application of heuristic algorithms in this area.

2.1. Electricity Emergency Distribution and Drone Distribution, Drone Fight Trajectory Issues

A timely provision of emergency supplies is necessary in the case of a sudden power outage to minimize the loss of lives and property caused by the outage and to restore the living and working environment. Hou et al. [2] proposed an electrical emergency rescue transport model based on node-integrated weights for resource satisfaction, creating an objective function with the least amount of material supply and time leanness. A multi-point fault repair optimization model was created by Gao et al. [14], considering the distribution of first aid materials, the cooperation of repair teams, and the order of fixing broken equipment.

However, traditional electrical material rescue is primarily found in urban locations, making it more practical in favorable traffic and road conditions. The drawbacks of the conventional method are more severe in areas with poor road conditions, making drone transportation a more effective means to increase transportation effectiveness. Below are articles related to drone transport.

By modeling the power consumption of a drone as a nonlinear function of payload and trip time in a multi-traveling Drone routing problem model, considering a time window, Cheng et al. [15] further extended Dorling's study. Their model offers logical and sub-gradient cuts to handle nonlinear power functions, using the branch-and-cut method to resolve the drone routing issue. In Gentili et al.'s [6] optimization issue, the value of emergency medical supplies is minimized depending on their perishability. Considering the drone's battery life while transporting medical supplies, they hypothesize that each platform would only have one drone that could service one node at a time. The application of VRPD for heterogeneous drones is also investigated in several of the studies. The application of VRPD for heterogeneous drones is also investigated in several of the studies. For a heterogeneous fixed fleet drone routing problem, Chowdhury et al. [16] suggested a mixed integer linear programming model to resolve the cost of post-disaster inspections because of several aspects. Chen et al. [17] dealt with the drone fight trajectory planning problem for drones with different capabilities in a multi-area system. They initially created an algorithm to group regions into clusters, drawing inspiration from density-based clustering techniques. They then obtained approximations of the ideal point-to-point pathways for drones to carry out coverage tasks. In a different work by Chen et al. [18], they concentrated on the issue of coverage fight trajectory planning for heterogeneous drones. They suggested a method based on the Ant Colony System (ACS) to obtain enough drone pathways and cover every region thoroughly and effectively. According to the literature, the most significant barrier to drone distribution research is the constraint, which is one of the reasons why conventional vehicle route problem models are inapplicable to drone logistics.

Most previous studies have examined the issue from a drone-specific angle, ignoring the coordinated relationship between supply timeliness and drone scheduling, as well as target demand and the drones' safety considerations. The construction of a drone scheduling model focusing on delivering electrical materials is crucial because models for the emergency transport of power drones are still uncommon.

2.2. Algorithm for Solving UAV Task Assignment

In drone task assignments, heuristic algorithms are more frequently utilized [19]. Based on this, Wu et al. [20] perform the secondary selection operation of the genetic algorithm (GA) to improve the population diversity, and the secondary selection operation adopts the improved simulated annealing algorithm (SA) to solve the collaborative multitasking allocation problem more effectively. Han et al. [21] suggested a fuzzy elite strategy genetic algorithm to handle complicated issues. To increase its capacity to escape local optimal traps and hasten convergence, Wang et al. [22] suggested combining Simulated Annealing (SA) and Large Neighborhood Search (LNS) algorithms. Liu et al. [23] also proposed a collaborative optimization method combining GA and clustering methods to satisfy the task assignment of forest fires by drones. An upgraded cellular automaton (CA) and an optimal spanning tree technique were utilized by Li et al. [24] to build the path network and find the best routes between various endpoints. Zhang et al. [25] dynamically divided the particle swarm based on the particle mass and changed the topology of the algorithm. In addition, dynamic problems in tasking problems are common. Additionally, the PSO algorithm has been employed in additional combinations as a heuristic method: in their studies, Geng et al. [26] proposed a quantified particle swarm optimization algorithm for the task allocation problem of UAV clusters;. Shao et al. [27] proposed a hybrid strategy based on discrete particle swarm optimization for the many-to-one task planning problem in the case of a constructed a quantized particle swarm optimization technique. The simulated annealing algorithm is enhanced by Chen et al. [28] using a Levi distribution strategy, and it works well for both dynamic and static task assignment models. Yang et al. [29] use a distributed rational clustering algorithm for UAV clusters based on sensor networks and mobile information to improve the completion rate of UAV task assignments, providing a new distributed algorithm for task assignments.

According to the literature above, traditional heuristic algorithms cannot perform the dynamic adjustment process because they rely solely on the initial parameter setting, omitting the impact of the algorithm updating process on the parameters.

Using deep learning to optimize the known algorithms can improve the efficacy of traditional control methods [30]. This paper uses this idea, using the ability of reinforcement learning to dynamically adjust the parameters [13] in order to make up for the shortcomings of the traditional algorithms.

When viewed in conjunction with the papers above, it is clear that the power emergency drone transport model suggested in this paper can target the power emergency drone distribution problem. The proposed TD3PSO algorithm addresses the issue of heuristic algorithms that depend on the initial conditions and the stability of the solution while also enhancing the algorithm's efficiency.

3. System Modeling and Problem Statement

In this study, we explore using drones to restore the power system following a disaster rapidly. The number of clients each drone can supply, the order in which they are delivered, the cargo capacity, the flight time, and the flight's safety all impact how reliable the delivery is. This article focuses on minimizing scheduling losses and delivering high-value, high-demand, and high-speed goods with more dependability to the greatest extent possible to meet drone scheduling requirements.

Based on the abovementioned issue, the set of customer points is defined as $N = \{1, 2, 3, \dots, j\}$, and the set of drones is defined as $H = \{1, 2, 3, \dots, k\}$ in this study. The flight paths of many drones are viewed as parallel systems; the following is an assumption of the problem.

- 1. Each drone is unique and maintains a constant speed during flight regardless of the surrounding mountainous environment;
- 2. The takeoff, landing, and service time of each drone are jointly counted as service time;
- 3. The drones have a fixed service time for each customer;
- 4. The drone can serve multiple customers per takeoff within its carrying capacity;
- 5. The UAV uses the same amount of energy during takeoff and landing as it does during flight and think of it as a matter of distance;
- 6. The distribution center has enough drones to accommodate customers who use them for deliveries near the center;
- 7. This study does not consider drone charging;
- 8. Only considering the quantity of material needed, not the diversity of material requirements;

- 9. The drone's maximum flying range equals two-thirds of its loaded range;
- 10. The weights in the formula sum to 1.

This study breaks the problem down according to the computing method described above. The power emergency objective model and constraint extension are covered in the first section, and the computation of power priority using the entropy power technique is covered in the second section. The first two components make up the static framework for allocating UAV tasks. The third component involves reallocating the task in an emergency while considering its dynamic environment.

3.1. Models

Table 1 below displays the parameter description for the time window-based UAV delivery model.

Parametric Variable	Description of Variables	Parametric Variable	Description of Variables
x_h	Whether or not the h drone was used	y_{ih}	Does drone <i>h</i> serve <i>i</i>
S_i	Service hours at point i	x_{iph}	Whether the h drone reaches <i>p</i> from i
ω_{ih}	Waiting time of the h drone at node i	x _{oih}	Does the <i>h</i> drone arrive from the warehouse <i>i</i>
φ _{ih}	Missed time of the h drone at node i	x _{ioh}	Does the <i>h</i> drone return to the warehouse from <i>i</i>
C ₁	The weighting of drone numbers	L	Maximum time constraint
C ₂	Time-consumption function weights	ej	Earliest arrival time for client <i>j</i> to receive service
C3	Time-balanced weighting	lj	Latest arrival time for <i>a</i> client <i>i</i> to receive services
C ₄	The weighting of UAV safety factors	W _{max}	A maximum drone carrying capacity
Z1	Number of drones function	t_{ih}	Drone runtime
Z ₂	Consumption time function	ΔN	Change in the number of drone missions
Z ₃	Consumption time function	m_h	The total time spent by the drone
Z_4	Security risk function	ω_i	Quantity of material at point i

Table 1. Description of symbols appearing in the model.

$$\min Z = Z_1 + Z_2 + Z_3 + Z_4 \tag{1}$$

$$Z_1 = C_1 \sum_{h=1}^{k} x_h$$
 (2)

$$Z_{2} = C_{2} \left(\sum_{i=1}^{j} \sum_{h=1}^{k} \varphi_{ih} + \sum_{i=1}^{j} \sum_{h=1}^{k} \omega_{ih} + \sum_{i=1}^{j} \sum_{h=1}^{k} t_{ih} + \sum_{h=1}^{k} S_{h} \right)$$
(3)

$$Z_{3} = C_{3}(\sqrt{\frac{\sum_{i=1}^{j}(m_{h} - \overline{m_{h}})^{2}}{j}})$$
(4)

$$Z_4 = C_4 \prod_{i=1}^{j} \sum_{n=1}^{k} \frac{1 - t_{ih} * v}{t_{ih} * v}$$
(5)

.s.t:

$$\sum_{i=1}^{j} \sum_{h=1}^{k} x_{iph} = 1, \forall j \in N$$
(6)

$$e_j \sum_{i=1}^j x_{iph} \le \varphi_{ih} + t_{ih} + \omega_{ih} \le l_j \sum_{i=1}^j x_{iph}$$

$$\tag{7}$$

$$\prod_{i=1}^{j} x_{ioh} = 1 \tag{8}$$

$$\sum_{k=1}^{m} \sum_{h=1}^{n} x_{oih} + \sum_{k=1}^{m} \sum_{h=1}^{n} x_{iph} = N$$
(9)

$$\sum_{i=0}^{j} \omega_i y_{ik} \le W_{max}, \forall k \in H$$
(10)

$$t_{ik} \le L, \forall j \in N, \forall k \in H \tag{11}$$

Equation (1) denotes the overall objective function. Equation (2) represents the number of drones needed to complete the mission. Equation (3) depicts the drone k-point task time function, which includes the flight time and service time functions and the actual time penalty function of the drone at each point. The drone time equilibrium function is shown in Equation (4). Equation (5) defines the drone flight risk function; where the more service stations and longer the drone path, the more expensive it is. Equation (6) shows that only a drone may visit the client demand points. The drone service time window must satisfy the conditions, according to Equation (7). Equation (8) states that the drone must resume flight. Equation (9) represents the requirements for completing all goal points. Equation (10) says that the drone load must weigh less than the drone flight object. Equation (11) shows that the drone must finish the task within the allotted time window.

3.2. Entropy Power Method to Calculate Power Distribution Priority

The Entropy Weight Method (EWM) may accurately depict the data's inherent regularity and informativeness through the entropy generated from the data. The EWM is the The calculation method is as follows, assuming that there are n evaluation items and m evaluation indicators:

Calculate the information entropy E_i for each indicator using the following Equation (12):

$$E_{j} = -\sum_{j=1}^{m} p_{ij} \ln p_{ij}$$
(12)

where P_{ij} is the outcome of normalizing the raw data obtained as Equations (13)–(15), displaying the following formula:

$$P_{ij} = \frac{x_{ij}}{\sum_{k=1}^{m} x_{ik}}, j = 1, 2, \dots, m$$
(13)

$$g_j = 1 - \frac{1}{\ln n} E_j \tag{14}$$

$$w_j = \frac{g_j}{\sum_{k=1}^m g_j} \tag{15}$$

The node charge coupling at the distribution point [2], the voltage fluctuation rate, the charge loss, the material requirement, and the power node level are the influencing elements considered in this work, which employs the entropy weight approach to rank the power system data.

3.3. Reassignment

The failure of drone task assignment can occur in various unforeseen circumstances; the failure situation considered in this paper is the new task point or drone damage. Depending on the effects of the failure situation, the drone task reallocation into global redistribution and local adjustment. The amount of task transformation threshold setting is 10 percent of the original task as Equation (16):

$$T = N/10 \tag{16}$$

$$\begin{cases} if \ \Delta N > T, \ global \ redistribution \\ if \ \Delta N \le T, \ Local \ adjustments \end{cases}$$
(17)

A full re-assignment of the mission is necessary if several change target points $(\Delta N > T)$ influence the mission execution for most drones. The mission data recording and complete reallocation are the two components of the complete reallocation flow. Similar to the job assignment process described above, the reassignment procedure involves changing the task assignment model from a homogeneous model to a heterogeneous drone one.

Local task allocation is initiated if there are few altered target points ($\Delta N \leq T$), and the effect on the overall task allocation is minimal. Under local task assignment, the current coordinates of the drones and uncompleted tasks are first solved for the coordinate. Next, the center of each drone's task assignment coordinates is calculated. Finally, the distance to the center of the new task point is compared to assigning the task and carrying out task sequencing. Figure 2 is a graphical representation of the task reassignment process described above.



Figure 2. Task reassignment flowchart.

4. Algorithm Design

4.1. PSO Optimization Algorithm

When allocating tasks to drones, one of the goal optimization algorithms known as the particle swarm algorithm is frequently used; the algorithm's main idea is to use information sharing between members of the swarm to cause the movement of the entire multitude to evolve from disorder to order in the problem solution space, leading to the discovery of a workable solution to the problem. The equations for its state update are provided in (18) and (19); the specific flow of the algorithm is shown in Algorithm 1.

$$v_i^{d+1} = \omega v_i^d + c_1 r_1 \left(pbest_i - x_i^d \right) + c_2 r_2 \left(gbest - x_i^d \right)$$
(18)

$$x_i^{d+1} = x_i^t + v_i^{d+1} (19)$$

Set the values of the parameters initialize the positions and the velocities of the particles Evaluate the fitness values, set X to be Pi and find Pg
t = 0;
While t < maxiter:
Update the velocity Vi of the particle by using Equation (16)
Update the position Xi of the particle by using Equation (17);
Update Pi and Pg;
t = t + 1;
End while.

4.2. Encoding and Decoding Process

It is necessary first to perform the discrete "coding" of the particles because the VRPD problem is discrete. Since the positions and velocities of the particles in the PSO algorithm are continuous variables and the corresponding solution space is also constant, it is impossible to decode the feasible solution for the task assignment directly.

N stands for the number of tasks, H for the number of drones, and the number of tasks is the particle dimension. Limit the size of the particle's position to [0, N] and the update velocity to [-1, 1] using real number encoding. Let the particle's position and velocity be N-dimensional real vectors. Then, in the decoding process, let Int(n) be the integer part of the actual number n, and P(N) denotes the fractional part of n, e.g., Int (3.56) = 3, p (3.65) = 0.56 where Int(n) = k indicates that the nth task is assigned to drone k. The size of the fractional part and the power node importance weight in the decoding process together determine the drone distribution task order, from which the task sequence Δ_T can be solved. This paper stipulate that the drone material transport is strictly assigned according to the weight parameters. In Figure 2 or in Table 2, the mission priority is displayed as weight sorting (a > b > c > d > e), and the mission decoding value is displayed as n, presuming there are three UAVs and five mission target points. Figure 3 shows the decoding procedure and outcomes.

Task	1	2	3	4	5
Weight Sorting	А	В	С	D	E
n	2.15	1.35	3.21	3.32	1.15

Table 2. value of task decoding.



Figure 3. Task decoding block diagram.

4.3. TD3 Algorithm

The TD3 algorithm is appropriate for handling multi-dimensional problems with continuous action spaces. In addition, it can effectively solve the problem of over-estimation of value Q, which is easy to appear in the DDPG algorithm, making the training process more stable. The specific flow of the algorithm is shown in Algorithm 2. The TD3 algorithm chooses operations based on the current policy and exploration noise, where \emptyset denotes the actor network's parameter settings, and a_{min} and a_{max} are the action space's lower and upper bounds, respectively, with a_{min} and a_{max} adhering to a Gaussian distribution:

$$a_t = clip(\pi_{\varnothing}(s_t) + \varepsilon, a_{min}, a_{max})$$
(20)

The TD3 algorithm uses an experience playback mechanism, where the agent enters trial data into a pool of experience buffers R and then randomly selects N batches of data from R to process the training network and modify its parameters. The TD3 approach uses two sets of critic networks to represent various Q-values to address overestimating values. The target Q-value is then estimated using the formula below.

$$y_i = r_i + \gamma \min Q_{\theta_k}(s_{i+1}, \tilde{a}_{i+1}) \tag{21}$$

where \tilde{a}_{i+1} is set as follows, similar to Equation (20):

$$\widetilde{a}_{i+1} = clip(\pi_{\varnothing}(s_t) + \varepsilon, a_{min}, a_{max})$$
(22)

The TD3 algorithm uses the minimization of the time-difference (TD) error to update the critic network parameters θ_k as follows:

$$\theta_k \leftarrow \arg_{\theta_k} \min \frac{1}{N} \sum_i \left(\left(y_i - Q_{\theta_1}(s_i, a_i) \right) \right)^2, k \in \{1, 2\}$$
(23)

The TD3 algorithm Q stabilizes when it does not update the Actor-network, lowering the number of incorrect updates and increasing algorithm stability. As a result, the TD3 algorithm updates the participant network a little less frequently than the critic-network. The TD3 method updates the participant network parameters using deterministic policy gradient, and the following equation displays the critic-network parameters.

$$\nabla_{\varnothing} j(\pi_{\varnothing}) \approx \frac{1}{N} \sum_{i} \nabla_{a} Q_{\theta_{1}}(s_{i}, a) \big|_{a=\pi_{\varnothing}(s_{i})} \nabla_{\varnothing} \pi_{\varnothing}(s_{i})$$
(24)

In addition, to ensure the stability of neural network training, a delayed update strategy is used:

$$\begin{cases} \theta_k \leftarrow \mu \theta_k + (1-\mu)\theta_k, k \in \{1,2\} \\ \dot{\varnothing} \leftarrow \mu \varnothing + (1-\mu)\dot{\varnothing} \\ 0 < \mu < 1 \end{cases}$$
(25)

Algorithm 2 TD3 algorithm

```
Initialize critic networks Q_{\theta_1}, Q_{\theta_2}, and actor network \pi_{\varnothing}
with random parameters \theta_1, \theta_2, \varnothing
Initialize target networks \hat{\theta_1} \leftarrow \theta_1 \ \hat{\theta_2} \leftarrow \theta_2, \emptyset \leftarrow \emptyset
Initialize replay buffer B
       for t = 1 to T do
                  Select action with exploration noise a_i \sim \pi_{\emptyset}(s) + \varepsilon,
                  \epsilon \sim N(0, \sigma) and observe reward r and new state ś
                  Store transition tuple (s, a, r, \hat{s}) in B
                  Sample mini-batch of K transitions(s, a, r, \dot{s}) form
                  \widetilde{a}_{i+1} = clip(\pi_{\varnothing}(s_t) + \varepsilon, a_{min}, a_{max})
         y_i = r_i + \gamma minQ_{\theta_k}(s_{i+1}, \tilde{a}_{i+1})
         Update critic \theta_k \leftarrow arg_{\theta_k}min\frac{1}{N}\sum_i ((y_i - Q_{\theta_1}(s_i, a_i)))^2, k \in \{1, 2\}
                 if t mod d then
                         \nabla_{\varnothing} j(\pi_{\varnothing}) \approx \frac{1}{N} \sum_{i} \nabla_{a} Q_{\theta_{1}}(s_{i}, a) \big|_{a = \pi_{\varnothing}(s_{i})} \nabla_{\varnothing} \pi_{\varnothing}(s_{i})
                         Update target networks:
\theta_k \leftarrow \mu \theta_k + (1-\mu)\theta_k, k \in \{1,2\}
\mathscr{O} \leftarrow \mu \mathscr{O} + (1 - \mu) \mathscr{O}
   end if
end for
```

4.4. TD3PSO Algorithm

It is easy for the PSO algorithm to fall into the defect of local optimization and the complexity of the model. In this paper, we propose an algorithm to optimize PSO parameters by combining TD3 dynamic parameterization strategy (TD3PSO), which can improve the algorithm accuracy, reduce the randomness of the algorithm, and improve the efficiency of UAV scheduling under the complex model.

Figure 4 below shows the algorithm flowchart made for this study by the TD3 algorithm update process and PSO algorithm update method mentioned before. The training flowchart is shown in the red dotted box; the specific process and the above TD3 algorithm process are similar. The specific flow of the TD3PSO algorithm is explained as shown in Algorithm 3. The TD3PSO algorithm running flow sketch is in the blue dotted box, as seen in the image. The state, action, and reward [13] settings for the particle swarm algorithm employed in this study are displayed below:

Algorithm 3 TD3PSO-trian algorithm

Initialize the number of population particles J, the maximum number of iterations T and the initial
parameters (<i>w</i> , <i>c</i> 1, <i>c</i> 2)
For I = 1:J
Initialize the particle location <i>Xi</i> and velocities <i>Vi</i>
end for
Initialize critic networks Q_{θ_1} , Q_{θ_2} , and actor network π_{\varnothing}
with random parameters $\theta_1, \theta_2, \emptyset$
Initialize target networks $\dot{\theta_1} \leftarrow \theta_1 \ \dot{\theta_2} \leftarrow \theta_2$, $\dot{\varnothing} \leftarrow \varnothing$
Initialize replay buffer <i>B</i>
for $t = 1$ to T do:
Obtain the state <i>state</i> from the pso
Form actor network to get action
Calculate the parameters w , $c1$ and $c2$
Update the particle location X and velocities V
Select action with exploration noise $a_i \sim \pi_{\emptyset}(s) + \varepsilon_i$
$\epsilon \sim N(0,\sigma)$ and observe reward r and new state \dot{s}
Store transition tuple (s, a, r, δ) in B

Algorithm 3 Cont.

Sample mini-batch of K transitions (s, a, r, δ) form $\widetilde{a}_{i+1} = clip(\pi_{\varnothing}(s_t) + \varepsilon, a_{min}, a_{max})$ $y_i = r_i + \gamma minQ_{\theta_k}(s_{i+1}, \widetilde{a}_{i+1})$ Update critic $\theta_k \leftarrow arg_{\theta_k}min\frac{1}{N}\sum_i ((y_i - Q_{\theta_1}(s_i, a_i)))^2, k \in \{1, 2\}$ if t mod d then Update & by the deterministic policy gradient: $\nabla_{\varnothing} j(\pi_{\varnothing}) \approx \frac{1}{N}\sum_i \nabla_a Q_{\theta_1}(s_i, a)|_{a=\pi_{\varnothing}(s_i)} \nabla_{\varnothing} \pi_{\varnothing}(s_i)$ Update target networks: $\theta'_k \leftarrow \mu \theta_k + (1 - \mu) \theta'_k, k \in \{1, 2\}, \& \leftarrow \mu \varnothing + (1 - \mu) \&$ end if end for



Figure 4. TD3PSO operation flowchart.

1. state:

The slight difference in the data change between 0 and 1 will be used to amplify the activation by applying the sin function activation, where $I = \frac{item}{item_{max}}$ in the equation above stands for the current iteration number, $D = var(x_{ij})$ for the level of particle dispersion, and $F = \frac{item_{noimpove}}{item_{max}}$ for the time when the particle has not been updated.

2. action:

According to the Table 3 below, the action set up for this work is a twenty-dimensional action space:

$$T \begin{cases} w = a[0] * 0.8 + 0.1\\ scale = \frac{1}{a[1] + a[2] + 0.0001} * a[3] * 8\\ c1 = scale * a[1]\\ c2 = scale * a[21]\\ c3 = scale * a[3] \end{cases}$$
(26)

3. Reward:

When the adaptation value derived from the current PSO computation is superior to that from the previous iteration, the reward value is 1.

$$\mathbf{R} = \begin{cases} 1, if best(t+1) > best(t) \\ -1, if best(t+1) < best(t) \end{cases}$$
(27)

Table 3. Action table.

Action	0	1	2	3	4	5	6	•••	19
А	a [0]	a [1]	a [2]	a [3]	a [0]	a [1]	a [2]	•••	a [3]

This paper's actor-network structure and critic-network structure are applied to both actors and critics in the target network, shown in Figure 5. The actor-network is mainly used in the training process structure and has two layers of fully linked layer and one layer of activation layer tanh function layer composition, the input of the state dimension of 15 dimensions, the output of the actor is 20 dimensions, the critical network mainly consists of five layers of fully linked layer and one layer of activation layer tanh function composition, the input for the state and the actor of the splicing into the output of the Q-value.



Figure 5. Actor-network and critic-network.

The TD3 algorithm can retrain the algorithm model by retaining data from the training process, such as network parameters and empirical sample pools. The corrected TD3 strategy network is stored as a parameter file and called by the PSO algorithm as an offline neural network function. The PSO algorithm speed update formula is presented below, and the pseudo-code for the algorithm is shown in Algorithm 4 below. The entire network design of the PSO algorithm based on this strategy is illustrated in Figure 4 Run section:

$$v_i^{d+1} = \omega v_i^d + c_1 r_1 \left(pbest_i - x_i^d \right) + c_2 r_2 \left(gbest - x_i^d \right) + c_3 r_{13} \left(pbest_{fi} - x_i^d \right)$$
(28)

Algorithm 4 TD3PSO algorithm
Procedure TD3PSO
Initialize the particle swarm and parameters (X, V)
While not stop
For I =1 to N
Decode current function, calculation state {(26), (27)}
Calculation action according to state and network
Update the V and X according to action
End for
End while
End procedure

5. Experiment

5.1. Simulation Experiments

This paper uses the 25-coordinate-point VRPTW arithmetic example from the traditional Solomon data set [31] of VRPTW to validate and analyze the task above the assignment model in an efficient simulation environment. This is implemented to verify the efficacy of the model proposed in this paper and the TD3PSO algorithm solution. In the numerical aspect, the FITNESS value in a single experiment and the average value in several experiments were used to compare this paper's algorithm and the effect of the algorithm in solving comparison problems. At the same time, the average difference of several experiments was used to compare the algorithm's stability. In the graphical aspect, the observation of solving the Gantt chart up to the time and the number of tasks of a single UAV, and the all-encompassing results were also used to compare the algorithm.

In this paper, c101, r101, and rc101 data are selected for analysis, where c101 represents the target point as a centralized target point, r101 represents the discrete target point, and rc101 represents the centralized-discrete target to solve the model effect and algorithmic solution effect under different types of target points. The details of the specific parameters can be found in the link in the appendix.

5.1.1. Parameterization

The Solomon data set and the parameters of common drones are used to set the settings for this experiment by the requirements of this paper, as shown in Table 4 below:

Parameter	Data Size	Parameter	Data Size
Number of drones	10	W1	0.13
Number of target points	25	W2	0.51
Maximum range of the drone	200	W3	0.10
Maximum drone payload	120	W4	0.26
Drone speed	5		

Table 4. Drone parameter settings.

5.1.2. PSO Algorithm Parameter Settings

The parameters of the simulated traditional PSO algorithm are established for the experiments in this work by the specifications, as stated in Table 5 below:

Tuble 5. 1 50 pulumeter settings	Tabl	e 5.	PSO	parameter	settings
---	------	------	-----	-----------	----------

Parameter	Data Size	Parameter	Data Size
population size	10	Initial particles C1, C2	2
number of particles	25	Update speed range	[0,1]
Initial weight	2	Updating the location interval	[0,25]

5.1.3. Simulation Results

Ten drones are employed in the scenario below to rescue twenty-five power-affected locations with equal levels of power importance at the mission sites. The table above displays the drones' and target points' initial settings. The maximum number of function evaluations, Fmax, is utilized as the algorithm iteration termination condition, and timemax = 1000 because different algorithms evaluate functions in varying numbers throughout each iteration. Each algorithm is run independently 20 times to lessen the impact of unpredictability in the algorithms.

Simulation Experiment 1: The c101 data set is used for simulation with the models and algorithms presented in this research.

The performance of iterative optimization in the data set c101 is better than the contrasting algorithm in this paper, as can be seen from experiment 1, which shows that the algorithm in this paper. As shown in Figure 6, after 1000 iterations, obtained the minimum fitness value of 13.675. However, the algorithm's iteration slows after 100 iterations, compared to the contrasting algorithms in the early stage of the fastest convergence, shown in this paper. Figures 7 and 8, respectively, exhibit the task allocation results and a Gantt chart. Figure 7 primarily shows the path display of the task allocation results, demonstrating how well the algorithmic method worked. Figure 8e UAVs take more time on average than with the Figure 8b,c enhancement, with the Figure 8a and the Figure 8d being near, while the maximum UAV flight duration is more centralized with the Figure 8a,b,d and equivalent to the function of the 113. Conclusion: The drone algorithm used in this paper has a particular advantage in job equalization, and the advantage in drone count is more pronounced. In addition, Table 6 shows that after 20 trials, the TD3PSO algorithm's standard deviation value is 3.6123, which is 8.32% lower than the RLPSO algorithm. This indicates that the algorithm used in this paper has the best stability when solving the task allocation model, and its average fitness value is 15.6335, which is 33.3% higher than the RLPSO algorithm.

Table 6. c101 results of 20 experiments.

Algorithm	Optimum	Average	Minimum	Standard Deviation	Whether There Are Unsolved Cases
PSO	28.8198	47.991	Inf	16.5067	YES
PSO-	42.4976	59.8670	Inf	16.7742	YES
MGRR-PSO	46.2517	59.6568	76.6045	12.4412	NO
RLPSO	17.8533	20.8414/33.3% ↑	32.5254	3.9125/8.32% ↑	NO
TD3PSO	8.3756	15.6335	21.9936	3.6123	NO



Figure 6. Convergence curve of the algorithm under c101 data set.



Figure 7. c101 data set task allocation map. (**a**): PSO task solving; (**b**): PSO- task solving; (**c**): MGRR-PSO task solving; (**d**): RLPSO task solving; (**e**): TD3PSO task solving. Different colors represent different UAVs and are apply to all task allocation maps in this paper.



Figure 8. c101 data set task allocation Gantt chart. (a): PSO task solving; (b): PSO- task solving; (c): MGRR-PSO task solving; (d): RLPSO task solving; (e): TD3PSO task solving.

Simulation Experiment 2: The r101 data set is used for simulation with the models and algorithms presented in this research.

The performance of iterative optimization in the data set r101 is better than the comparison algorithms in this paper, as seen in experiment 2. The algorithm in this paper obtained the minimum fitness value of 5.8375 after 1000 iterations, indicating that the iterative process of convergence, in general, is successful because the PSO algorithm is added in the prevention of falling into the local optimum component. The paths in the outcomes of the task assignment are primarily shown in Figure 9, demonstrating the efficiency of the algorithmic solution. The task allocation result graph and Gantt chart are displayed in Figures 10 and 11. From Figure 10, it is clear that Figure 10e UAVs are in the minimum number of 6, the UAV's maximum flight time is concentrated between 130 and 176, and Figure 11a–d is comparable to Figure 11a–d. The average time consumed by UAVs has improved significantly over Figure 11a-c compared to Figure 11d in some ways. This leads one to conclude that the drone algorithm presented in this research has a distinct advantage in task distribution, with the number of drones being more pronounced. As shown in Table 7, the mean value of fitness is 5.7227, which is 24.1% higher than the RLPSO algorithm, and the standard deviation value of the TD3PSO algorithm is 3.6123, reduced by 47.4% compared to the RLPSO algorithm, showing that this paper's algorithm has the best stability in the process of solving the task allocation model. The findings demonstrate that the algorithm can solve the UAV task distribution problem using the r101 data set.



Figure 9. Convergence curve of the algorithm under c101 data set.



Figure 10. r101 data set task allocation map. (**a**): PSO task solving; (**b**): PSO- task solving; (**c**): MGRR-PSO task solving; (**d**): RLPSO task solving; (**e**): TD3PSO task solving.



Figure 11. r101 data set task allocation Gantt chart. (a): PSO task solving; (b): PSO- task solving; (c): MGRR-PSO task solving; (d): RLPSO task solving; (e): TD3PSO task solving.

Algorithm	Optimum	Average	Minimum	Standard Deviation	Whether There Are Unsolved Cases
PSO	9.501	13.891	Inf	4.1239	Yes
PSO-	11.1921	14.0694	Inf	2.9179	Yes
MGRRPSO	12.0866	16.4385	Inf	2.5426	Yes
RLPSO	5.9323/	7.1032/24.1% ↑	9.2196/	$0.8214/47.4\%\uparrow$	No
TD3PSO	4.9821	5.7227	6.8297	0.5554	No

Fable 7. r101 Results of 20 experime	nts
---	-----

Simulation Experiment 3: The rc101 data set is used for simulation with the models and algorithms presented in this research.

As can be seen from experiment 3, the algorithm in this paper obtained the smallest fitness value of 6.3575 after 1000 iterations compared to other algorithms, indicating that the performance of iterative optimization in the data set rc101 is better than the comparison algorithm in this paper. After 70 iterations, the algorithm's iteration slows down, but preconvergence is quick compared to the comparison algorithm, as shown in Figure 12. The task allocation results and Gantt chart are shown in Figures 13 and 14. Figure 13 primarily displays the path in the task allocation results, demonstrating the efficacy of the algorithm solution; from Figure 14, we can see that Figure 14e UAVs have the fewest advantages (8), have the shortest maximum flight times, and have the most significant average times when compared to Figure 14a–d. This leads one to conclude that the number of drones used in this paper's algorithm is less of a benefit than the job balance. After 20 trials, the TD3PSO algorithm's standard deviation is 0.6197, 22.6% lower than the RLPSO algorithm's value, As shown in Table 8. This indicates that the algorithm used in this paper reduces the stability of the process for solving the task allocation model. The average fitness value is 6.1587,

which is 21.6% higher than the RLPSO algorithm's value, and the algorithm did not fail to solve the situation.

Algorithm	Optimum	Average	Minimum	Standard Deviation	Whether There Are Unsolved Cases
PSO	10.4205	17.0155	Inf	3.263	YES
PSO-	10.5521	15.2532	Inf	3.3360	YES
MGRRPSO	10.5118	16.1179	20.8634	2.7259	NO
RLPSO	6.5764	7.4918/21.6% ↑	8.2932	0.4798/22.6%↓	NO
TD3PSO	5.3194	6.1587	7.538	0.6197	NO

Table 8. rc101 results of 20 experiments.



Figure 12. Convergence curve of the algorithm under rc101 data set.



Figure 13. rc101 data set task allocation map. (**a**): PSO task solving; (**b**): PSO- task solving; (**c**): MGRR-PSO task solving; (**d**): RLPSO task solving; (**e**): TD3PSO task solving.



Figure 14. rc101 data set task allocation Gantt chart. (a): PSO task solving; (b): PSO- task solving; (c): MGRR-PSO task solving; (d): RLPSO task solving; (e): TD3PSO task solving.

The abovementioned simulation shows that this study outperforms the conventional method regarding the algorithm's problem-solving ability. The solving impact of this work is, on average, 28.6% better than the RLPSO algorithm, and the algorithm balance is 12.1% better. The task duration is shorter, the task balance is better, and the number of UAVs is smaller. This paper's algorithm can provide a better allocation scheme for solving the task allocation problem under the UAV transportation model. At the same time, the reduced volatility shows that the algorithm will not fall into the local optimum in solving the problem. Under different types of data sets, the TD3PSO algorithm can be more effective and have lower volatility than RLPSO. In addition, numerous experimental proofs support the viability of the task model presented in this study.

5.2. Real Scene Simulation Experiment

This paper simulates a genuine situation utilizing the data [32] and the IEEE33 data set to further test the model's efficacy and the TD3PSO algorithm solution described in this paper is used to carry out the actual scenario application. The following accurately describes the situation: a geological disaster in Wenchuan County in July 2012 paralyzed a portion of the electrical grid, necessitating the shipment of supplies for the necessary repairs.

Data assumptions are created by the requirements of this paper, taking into account the distribution capacity and demand for big UAVs. The experimental data assumptions are provided in Table 9 below, and Figure 15 displays their accurate coordinates and two-dimensional coordinate points.

Parameter	Data Size	Parameter	Data Size
Number of drones	6	Maximum drone payload	100
Number of target points	10	Drone speed	40
Maximum range of the drone	2800		
		UAV/target position	n target 9 0 300 350
	<u>e</u>	X	
(a)		(D)	

Table 9. Drone parameter settings.

Figure 15. Coordinates of affected points. (a): true coordinates; (b): emulation coordinate.

The distribution target points in this study are displayed in Table 10 below. Node significance is arranged with the findings of the entropy weighting technique, which are shown in Tables 10 and 11, and the jobs are allocated in the decoding process strictly in line with the sorting criteria.

The different solution results of the weights will also be different. The weights' size indicates that this mission's focus is different. To eliminate the influence of the weights on the algorithm of this paper, the following experiments are proposed. The experiments described in this paper were performed to eliminate the influence of weights on the algorithm. First, weight data were randomly generated into groups, each averaging 20 operations. One of the five experimental results is shown in Table 12 below.

Table 10.	Experimental	data setting a	and weighting	results.
-----------	--------------	----------------	---------------	----------

Disaster Area	Nodal Charge Coupling	Rate of Voltage Change	Electric Charge Loss	Material Requirements	Node Type	Importance Ranking
А	0.0570	0.4651	20	1500	0.3	7
В	0.06475	0.2453	90	350	0.1	9
С	0.0467	0.3125	30	330	0.6	2
D	0.0646	0.2345	12	390	0.6	3
E	0.0533	0.3412	16	863	0.1	5
F	0.0604	0.1545	67	125	0.3	8
G	0.0669	0.2347	43	41	0.1	6
Н	0.0594	0.4437	12	102	0.1	1
I	0.0583	0.1564	17	60	0.3	4
J	0.0583	0.4521	12	68	0.3	0

Disaster Area	Geographic Information Coordinates	Service Time	Time Window	x	у
А	[103.417331, 31.308881]	0.2	[1.5-6]	150	260
В	[103.544643, 31.519391]	0.3	[3.5-6.5]	41	540
С	[103.373051, 31.313721]	0.6	[1-4]	147	162
D	[103.612874, 31.390961]	0.6	[0.5-6]	109	659
Е	[103.550031, 31.443341]	0.3	[3–9]	81	555
F	[103.739295, 31.529901]	0.3	[1-3.5]	37	976
G	[103.495751, 31.356101]	0.5	[0.5-4]	126	435
Н	[103.583054, 31.485275]	0.5	[2-6.5]	59	629
Ι	[103.440382, 30.941994]	0.2	[1-6.5]	340	310
J	[103.456212, 30.993813]	0.3	[0.5 - 4.5]	313	345
start	[103.629139, 31.010113]			730	309

Table 11. Experimental data setting and location.

Table 12. Randomized weighting experiment results.

Disaster Area	Geographic Information Coordinates	Service Time	Time Window	x	у
1	$W_1 = 0.18$ $W_2 = 0.65$ $W_3 = 0.11$ $W_4 = 0.06$	5.6531 (inf)	4.6146 (inf)	2.8495/11.6%↑	2.5491/1.1%↑
2	$W_1 = 0.23$ $W_2 = 0.50$ $W_3 = 0.20$ $W_4 = 0.07$	6.3538 (inf)	4.8595 (inf)	3.4807/14.9%↑	2.9079/2.6% ↑
3	$W_1 = 0.59$ $W_2 = 0.27$ $W_3 = 0.04$ $W_4 = 0.10$	7.0175 (inf)	5.6140 (inf)	4.6436/13.2%↑	4.1049/0.97% ↑
4	$W_1 = 0.01$ $W_2 = 0.88$ $W_3 = 0.18$ $W_4 = 0.01$	6.8008 (inf)	4.8783 (inf)	3.9415/17%↑	3.3766/2.7%↑
5	$W_1 = 0.21$ $W_2 = 0.20$ $W_3 = 0.52$ $W_4 = 0.07$	6.5637 (inf)	3.9390 (inf)	3.6351/ 5.4% ↑	3.4583/1.1%↑

In this experiment, it can be seen that the algorithms successfully find the optimal solution under various weight conditions superimposed on the priority case (inf represents the task allocation scheme with failure). The average values of the MGRRPSO and RLPSO algorithms are up by 12.42% and 1.69%, respectively, compared to the algorithms in this paper, which can be proved successful under various weight conditions.

The weights are set in this work to match those in the Table 3 in the preceding section to fit the actual circumstances, the simulation experiments are run for the real situation, and the results of the experiments are displayed in Figure 16 and Table 13 below.





UAV		PSO	PSO-	MGRRPSO	DDPGPSO	TD3PSO
	U1	8	8	9-6-4-1		7-2
	U2	7-6-3			6-5	
	U3	2-0	2		9-1	4-3-1
Distribution	U4		7-4-1	0	4-3	9-5
	U5	5-1	6-5	8-3-2	7-2-0	8
	U6	9-4	9-3-0	7-5	8	6-0
Fitness	5	3.2223/ 5.3%	3.2323/ 5.6%	3.2312/ 5.5%	3.1235 /2.0%	3.0610

Table 13. UAV task assignment results and fitness size.

Figures 17 and 18 below show the results of tasking and the Gantt chart.



Figure 17. Real experiment task allocation map. (a): PSO task solving; (b): PSO- task solving; (c): MGRR-PSO task solving; (d): RLPSO task solving; (e): TD3PSO task solving.



Figure 18. Real experiment task allocation Gantt chart. (**a**): PSO task solving; (**b**): PSO- task solving; (**c**): MGRR-PSO task solving; (**d**): RLPSO task solving; (**e**): TD3PSO task solving.

Based on the information in Table 14, the distance, material requirements, flight risk, time, and task balance data for each plan are calculated as follows, The experimental results show that the TD3PSO algorithm in real scenarios obtained the following results: the shortest distance is 7578.1174, the task is entirely distributed, the flight risk is slightly higher compared to the PSO algorithm, and the RLPSO algorithm is 3.3847—the least amount of time consumed for the 7.7078, which can be found in the case of the integrated objective function, and thus the algorithms of this paper can be effective in solving the real-world scenarios.

Table 14. Plot of results for each parameter of the algorithm run.

Algorithm	Distance	Material Requirements (Normalization)	Flight Risk	Time	Task Balance
PSO	7945.3258	2.5526	3.2906	7.9862	Good
PSO-	8231.2050	2.5526	3.4344	7.7265	Poor
MGRRPSO	7886.1778	2.5526	3.7837	9.3205	Very poor
RLPSO	7650.7650	2.5526	3.3020	8.1524	Very good.
TD3PSO	7578.1174 ↓	2.5526	3.3847 ↑	7.7078 ↓	Very good

5.3. Reassignment of Tasks

The UAV redistribution problem can be abstracted into a centralized allocation scheme for heterogeneous UAVs with a time window and a distributed local task re-scheduling scheme by the redistribution scheme proposed in this paper. The following experiments can validate the effectiveness of this paper's redistribution models and algorithms in solving the redistribution problem. This work selects task point addition and anomalous UAV damage as unexpected circumstances.

5.3.1. Complete Redistribution Experiment

This study simulates the above algorithm results at t = 2.2 moments, where Drone 2 is damaged and unable to proceed with the following task allocation scheme. At this point, the task is changed to a heterogeneous drone allocation task, and the current drone's location serves as the drone's new starting point for the task allocation calculations, as the Table 15. The current coordinate information is also provided.

Drone	Location	Information Note	Task	Current Material Quantity (Normalization)
Drone1	(291.957, 272.421)	Coordinates transformed	7-2	1.5
Drone 2	(730.4238, 309.1158)	No task was performed.	0	1.5
Drone 3	/	Implementation of completion Task 4	3-1	
Drone 4	(314, 451.212)	Coordinates transformed	9-5	1.5
Drone 5	(413.505, 614.341)	Coordinates transformed	8	1.5
Drone 6	(299.695, 398.962)	Coordinates transformed	6-0	1.5

Table 15. Moment of complete redistribution of drones.

As can be seen from Figure 19 and Table 16, this experiment can demonstrate that, in the case of complete redistribution, the redistribution scheme proposed in this paper is capable of effectively carrying out the task reallocation to provide dynamics for the electric power task allocation, and the algorithm in this paper is also capable of performing the dynamic complete redistribution.



Figure 19. Complete redistribution task allocation. (a): task allocation map; (b): Gantt chart.

Table 16. Program for the distribution of tasks.

Drone	Task of Drone	Fitness
Drone 1	5	
Drone 2	2-0	
Drone 4	9-7-1	1.8134
Drone 5	8-3	
Drone 6	6	

5.3.2. Local Redistribution Experiment

This study we model that a new mission needs to be replanned at time = 2.2, and the distance between the additional mission point and the drone coordinate mission set must be estimated to integrate this mission point into its closest target point. The coordinates of each drone mission's clustering centroid at the trigger are displayed in Table 17 below.

Drone	Location	Current Status of Drones	Mission Centers	Current Material Quantity (Normalization)
Drone 1	(291.957, 272.421)	No task was performed 7-2	(196.478, 231.693)	1.5
Drone 1	(730.423, 309.115)	No task was performed	(730.423, 309.115)	1.5
Drone 1	(85.529, 575.551)	Implementation of Completion Task 4	(84.954, 633.527)	1.27
Drone 1	(314, 451.212)	No task was performed 9-5	(145.871, 517.059)	1.5
Drone 1	(413.505, 614.341)	No task was performed 8	(225.265, 795.633)	1.5
Drone 1	(299.695, 398.962)	No task was performed 6-0	(246.659, 393.271)	1.5

Table 17. Moment of complete redistribution of drones.

Add task information as shown in Tables 18 and 19 below:

Table 18. New mission data (part 1).

Disaster Area	Nodal Charge Coupling	Rate of Voltage Change	Electric Charge Loss	Material Requirement	Node Type
К	0.0677	0.2934	15	300	0.3

Table 19. New mission data (part 2).

Disaster Area	Geographic Information Coordinates	Service Time	Time Window	x	у
К	[103.617281, 31.490174]	0.4	[2.2–5.5]	57	

The approach suggested in this study may successfully address the issue of local task assignment, provide a dynamic for emergency electricity material drone assignment, and enhance the holistic character of task assignment, as shown in Figure 20 and Table 20.



Figure 20. Complete Redistribution task allocation map (a) and Gantt chart (b).

Drone	Task
Drone 1	7-2
Drone 2	
Drone 3	4-/K-3-1
Drone 4	9-5
Drone 5	8
Drone 6	6-0

Table 20. Program for the distribution of tasks.

6. Discussion

The power emergency material dispatching model, its solution method, and the dynamic scenario are the areas of focus of this paper. These three areas are addressed below:

Task balance, power priority, and danger to UAV flight safety are all introduced while building the scheduling model. The influence of UAV flight on mission delivery owing to errors, environmental interference, and other factors can be successfully reduced by the flight safety risk function, and the mission balance can also effectively prevent UAV loss. Additionally, the power priority is introduced to offer objective parameters that can affect the particular power transportation problem; according to earlier research [5,6], most models do not consider such aspects. The findings demonstrate that the method in this study can address the issue raised in this paper.

The solution method of the scheduling model uses an adaptive parameter tuning approach, which combines reinforcement learning optimization heuristic algorithms [13,32]. Improved algorithms from earlier studies perform better and can tackle specific issues. This study introduces the TD3 algorithm, which is used to tune the parameters of the PSO algorithm. Experiments have demonstrated that it has a particular impact on the Solomon data set and real solving while also having a high level of stability. However, certain factors, including the weight value, the distribution of mission points, the number of UAVs, and the number of mission target points, will interfere with the method of this work in the actual tests. The mission point distribution and weight value have less impact on the way, as shown by the simulation and real-world experiments in this research. Further, because this work does not cover these, the number of UAVs and the number of mission target.

While building a framework for dynamic scheduling, some sudden problems can be successfully solved [27,28]. These solutions for dynamic scheduling can be divided into two categories: one, as in [29], introduces a distributed algorithmic framework to solve directly; and the other, as in this paper, presents subjective judgement conditions and categorizes the solution. The outcomes demonstrate the viability of the methodology used in this paper within the redistribution framework.

This essay has some restrictions as well. The contingencies included in the static model and task reassignment are presumptive events with little impact on the other scheduling scheme. Still, the task classification used in this paper's reassignment framework has some subjectivity that will affect the assignment effect and prevent task reassignment from keeping up with the situation. However, significant weather changes [33] can also impact UAV scheduling in real-world contexts. The trouble with the electricity system is likewise volatile. Such disruptive events will disrupt the dynamic scheduling plan. The scheduling plan will be impacted by aircraft problems, battery range, drug capacity, etc., and flight environment effects. This research does not analyze how aircraft operating circumstances affect the dispatching scheme because there are data gaps in the problem.

7. Conclusions

This paper develops a multi-drone collaborative task allocation model based on the target scenario of multi-drone multi-distribution of emergency power materials in mountainous areas. The model combines the entropy power method to find the priority of power nodes and the idea of dynamic allocation, which comprehensively solves the task allocation problem of power drones in the mountain and provides a viable solution for the transport of power emergency supplies. At the same time, the TD3PSO algorithm suggested in this study outperforms the conventional algorithm in terms of adaptability and stability and can address the drone job allocation problem.

Although the current work in this study primarily focuses on the power emergency distribution allocation problem, and its solution the drone flight trajectory problem also influences the allocation effect, the technique in this research can also be utilized for other task allocation problems. As a result, future studies will be crucial in focusing on the trajectory influence and increasing an algorithm's general usability.

Author Contributions: Conceptualization, methodology and writing-review and editing, W.Z.; software, validation, data curation and writing-original draft preparation, J.W.; Supervision, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Supported by National Social Science Foundation of China (19BGL120).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are openly available at. CervEdin/solomon-vrptw-benchmarks (github.com) (accessed on 16 March 2022).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

The following	abbreviations are used in this manuscript.
PSO	Particle swarm optimization algorithm
TD3PSO	Double-delay depth deterministic policy gradient particle swarm optimization algorithm
RLPSO	Reinforcement Learning Particle swarm optimization algorithm
EWM	Entropy Weight Method
PSO-	Roulette Particle swarm optimization algorithm
MGRRPSO	Multi-Group Particle swarm optimization algorithm
UAV	unmanned aerial vehicle
VRP	Vehicle Routing Problem
VRPTW	Vehicle routing problem with time window
VRPD	Vehicle Routing Problem of Drones
GA	Genetic Algorithm
DABC	Dynamical Artificial Bee Colony
ACS	Ant Colony System
CA	cellular automaton
SA	Simulated Annealing

References

- 1. Shen, Y.; Qian, T.; Li, W.; Zhao, W.; Tang, W.; Chen, X.; Yu, Z. Mobile energy storage systems with spatial-temporal flexibility for post-disaster recovery of power distribution systems: A bilevel optimization approach. *Energy* **2023**, *282*, 128300. [CrossRef]
- Hou, H.; Geng, H.; Xiao, X.; Huang, Y.; Yu, S.; Yu, J.F.; Tang, J.R. Research on dispatching model of electric emergency materials based on comprehensive weight of nodes. *Power Syst. Prot. Control* 2019, 47, 165–172.
- 3. He, Y.; Xiong, J.; Cheng, W.; Yang, J.; He, W.; Yong, Z.; Duan, Y.; Liu, J.; Yang, G.; Wang, N. Assessing the risk posed by flash floods to the transportation network in southwestern China. *Geocarto Int.* **2022**, *37*, 13210–13228. [CrossRef]
- 4. Shi, Y.; Lin, Y.; Li, B.; Li, R.Y.M. A bi-objective optimization model for the medical supplies' simultaneous pickup and delivery with drones. *Comput. Ind. Eng.* **2022**, 171, 108389. [CrossRef]
- Ghelichi, Z.; Gentili, M.; Mirchandani, P.B. Logistics for a fleet of drones for medical item delivery: A case study for Louisville, KY. Comput. Oper. Res. 2021, 135, 105443. [CrossRef]
- 6. Gentili, M.; Mirchandani, P.B.; Agnetis, A.; Ghelichi, Z. Locating platforms and scheduling a fleet of drones for emergency delivery of perishable items. *Comput. Ind. Eng.* 2022, *168*, 108057. [CrossRef]

- 7. Zhang, C.; Zhou, W.; Qin, W.; Tang, W. A novel UAV path planning approach: Heuristic crossing search and rescue optimization algorithm. *Expert Syst. Appl.* **2023**, *215*, 119243. [CrossRef]
- Zhang, A.; Xu, H.; Bi, W.; Xu, S. Adaptive mutant particle swarm optimization based precise cargo airdrop of unmanned aerial vehicles. *Appl. Soft Comput.* 2022, 130, 109657. [CrossRef]
- 9. Wang, X.; Liu, Z.; Li, X. Optimal delivery route planning for a fleet of heterogeneous drones: A rescheduling-based genetic algorithm approach. *Comput. Ind. Eng.* 2023, 179, 109179. [CrossRef]
- Lei, D.; Cui, Z.; Li, M. A dynamical artificial bee colony for vehicle routing problem with drones. *Eng. Appl. Artif. Intell.* 2022, 107, 104510. [CrossRef]
- Jain, M.; Saihjpal, V.; Singh, N.; Singh, S.B. An overview of variants and advancements of PSO algorithm. *Appl. Sci.* 2022, 12, 8392. [CrossRef]
- Suryanto, N.; Kang, H.; Kim, Y.; Yun, Y.; Larasati, H.T.; Kim, H. A distributed black-box adversarial attack based on multi-group particle swarm optimization. Sensors 2020, 20, 7158. [CrossRef]
- Yin, S.; Jin, M.; Lu, H.; Gong, G.; Mao, W.; Chen, G.; Li, W. Reinforcement-learning-based parameter adaptation method for particle swarm optimization. In *Complex & Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–25.
- 14. Gao, Z.L.; Xu, M.K.; Ding, S.Y.; Liu, Z.; Li, Y. Optimization scheduling of multi-fault rush repair for distribution networks based on modified artificial bee colony algorithm. *Power Syst. Prot. Control* **2019**, *13*, 107–114.
- 15. Cheng, C.; Adulyasak, Y.; Rousseau, L.M. Drone routing with energy function: Formulation and exact algorithm. *Transp. Res. Part B Methodol.* **2020**, *139*, 364–387. [CrossRef]
- Chowdhury, S.; Shahvari, O.; Marufuzzaman, M.; Li, X.; Bian, L. Drone routing and optimization for post-disaster inspection. *Comput. Ind. Eng.* 2021, 159, 107495. [CrossRef]
- 17. Chen, J.; Du, C.; Zhang, Y.; Han, P.; Wei, W. A clustering-based coverage path planning method for autonomous heterogeneous UAVs. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 25546–25556. [CrossRef]
- Chen, J.; Ling, F.; Zhang, Y.; You, T.; Liu, Y.; Du, X. Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. *Swarm Evol. Comput.* 2022, 69, 101005. [CrossRef]
- 19. Lalwani, S.; Sharma, H.; Satapathy, S.C.; Deep, K.; Bansal, J.C. A survey on parallel particle swarm optimization algorithms. *Arab. J. Sci. Eng.* **2019**, *44*, 2899–2923. [CrossRef]
- 20. Wu, X.; Yin, Y.; Xu, L.; Wu, X.; Meng, F.; Zhen, R. Multi-UAV task allocation based on improved genetic algorithm. *IEEE Access* **2021**, *9*, 100369–100379. [CrossRef]
- Han, S.; Fan, C.; Li, X.; Luo, X.; Liu, Z. A modified genetic algorithm for task assignment of heterogeneous unmanned aerial vehicle system. *Meas. Control* 2021, 54, 994–1014. [CrossRef]
- 22. Wang, X.; Zhou, J.; Yu, X.; Yu, X. A Hybrid Brain Storm Optimization Algorithm to Solve the Emergency Relief Routing Model. *Sustainability* **2023**, *15*, 8187. [CrossRef]
- Liu, X.; Jing, T.; Hou, L. An FW–GA Hybrid Algorithm Combined with Clustering for UAV Forest Fire Reconnaissance Task Assignment. *Mathematics* 2023, 11, 2400. [CrossRef]
- 24. Li, S.; Zhang, H.; Li, Z.; Liu, H. An air route network planning model of logistics UAV terminal distribution in urban low altitude airspace. *Sustainability* **2021**, *13*, 13079. [CrossRef]
- 25. Zhang, J.; Chen, Y.; Yang, Q.; Lu, Y.; Shi, G.; Wang, S.; Hu, J. Dynamic task allocation of multiple UAVs based on improved A-QCDPSO. *Electronics* 2022, *11*, 1028. [CrossRef]
- Geng, R.; Ji, R.; Zi, S. Research on task allocation of UAV cluster based on particle swarm quantization algorithm. *Math. Biosci.* Eng. 2022, 20, 18–33. [CrossRef]
- Shao, S.; Li, H.; Zhao, Y.; Wu, X. A New Method for Multi-UAV Cooperative Mission Planning Under Fault. *IEEE Access* 2023, 11, 52653–52667. [CrossRef]
- 28. Chen, C.; Li, Y.; Cao, G.; Zhang, J. Research on dynamic scheduling model of plant protection UAV based on levy simulated annealing algorithm. *Sustainability* **2023**, *15*, 1772. [CrossRef]
- 29. Yang, J.; Huang, X. A distributed algorithm for UAV cluster task assignment based on sensor network and mobile information. *Appl. Sci.* **2023**, *13*, 3705. [CrossRef]
- Heidari, A.; Navimipour, N.J.; Jamali, M.A.J.; Akbarpour, S. A Hybrid Approach for Latency and Battery Lifetime Optimization in IoT Devices through Offloading and CNN Learning. *Sustain. Comput. Inform. Syst.* 2023, 39, 100899. [CrossRef]
- 31. Xing, Y.Q. Research on Optimization of Distribution Path of Power Grid Emergency Supplies under Power IOT. Master's Thesis, North China Electric Power University, Beijing, China, 2021.
- 32. Wu, D.; Wang, G.G. Employing reinforcement learning to enhance particle swarm optimization methods. *Eng. Optim.* 2022, 54, 329–348. [CrossRef]
- Song, Y.; Zhang, B.; Wang, J.; Kwek, K. The impact of climate change on China's agricultural green total factor productivity. *Technol. Forecast. Soc. Chang.* 2022, 185, 122054. [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.