applied sciences

MDPI

Article

A Discrete Particle Swarm Optimization Algorithm for Dynamic Scheduling of Transmission Tasks

Xinzhe Wang [1,2,*] and Wenbin Yao [1]

[1] College of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China; yaowenbin_cdc@163.com
[2] Radio and Television & New Media Intelligent Monitoring Key Laboratory of NRTA (Radio, Film & Television Design & Research Institute), Beijing 100045, China
* Correspondence: wangxinzhe@drft.com.cn

Abstract: The dynamic-scheduling problem of transmission tasks (DSTT) is an important problem in the daily work of radio and television transmission stations. The transmission effect obtained by the greedy algorithm for task allocation is poor. In the case of more tasks and equipment and smaller time division, the precise algorithm cannot complete the calculation within an effective timeframe. In order to solve this problem, this paper proposes a discrete particle swarm optimization algorithm (DPSO), builds a DSTT mathematical model suitable for the DPSO, solves the problem that particle swarm operations are not easy to describe in discrete problems, and redefines the particle motion strategy and adds random disturbance operation in its probabilistic selection model to ensure the effectiveness of the algorithm. In the comparison experiment, the DPSO achieved much higher success rates than the greedy algorithm (GR) and the improved genetic algorithm (IGA). Finally, in the simulation experiment, the result data show that the accuracy of the DPSO outperforms that of the GR and IGA by up to 3.012295% and 0.11115%, respectively, and the efficiency of the DPSO outperforms that of the IGA by up to 69.246%.

Keywords: task dynamic scheduling; discrete particle swarm optimization algorithm; probability selection model; random disturbance; schedule; evaluation of transmission effect

Citation: Wang, X.; Yao, W. A Discrete Particle Swarm Optimization Algorithm for Dynamic Scheduling of Transmission Tasks. *Appl. Sci.* **2023**, *13*, 4353. https://doi.org/10.3390/app13074353

Academic Editor: Juan A. Gómez-Pulido

Received: 13 March 2023
Revised: 27 March 2023
Accepted: 28 March 2023
Published: 29 March 2023

1. Introduction

Radio- and television-transmitting stations usually deploy multiple equipment to complete multiple transmission tasks every day. Different equipment can be selected for each transmission task, but the effect of the execution on different equipment is different. On-duty personnel usually draw the transmission plan based on experience and control the equipment to complete the task according to the plan [1,2]. However, this does not guarantee the optimal transmission effect of the prepared plan. In addition, when a temporary task is added and the transmission plan needs to be adjusted, the transmission plan cannot be adjusted according to the transmission effect in time to obtain a new transmission plan with the best effect. To solve the above problems and improve the field of radio and television coverage, it is urgent to complete the dynamic scheduling of transmission tasks (DSTT) through intelligent algorithms and improve the transmission effect.

In previous research, we completed the static allocation of transmission tasks, that is, at the same time point, we assigned multiple tasks to multiple equipment with the goal of achieving an optimal comprehensive transmission effect [3]. The static-task-allocation problem can be summarized as a combinatorial optimization problem (COP), which is an NP-Hard problem. When there are fewer tasks and equipment, the enumeration algorithm is used to accurately calculate the global-optimal solution; when the number of tasks and equipment increases, the computing time of the enumeration algorithm increases exponentially. By introducing intelligent algorithms to solve this, the optimal solution can be calculated within an acceptable timeframe. Based on this, to calculate the task

allocation of multiple time periods, the problem can be broken down into independent allocations of multiple time periods. However, in practice, once the transmission task starts, the equipment cannot be interrupted and replaced before it ends. The calculation method of independent allocations gives priority to the previous tasks' optimal selection. The optimal allocation of the subsequent mission reduces the possibility of optimization, and the overall view is that the global-optimal solution cannot be obtained. Based on a global comprehensive evaluation, the scheduling problem of multi-period tasks for multiple equipment is thus solved thanks to the DSTT.

1.1. Related Works

This is a research paper written by Chinese radio and television researchers on the preparation of a transmission plan. In the process of preparing the plan, the greedy algorithm (GR) is used to find the priority allocation of the tasks that can obtain the optimal allocation of all tasks. According to this principle, until all tasks are scheduled, the number of algorithm iterations is related to the task period, and the backtracking algorithm is used to deal with conflicts in the scheduling. Although the algorithm has been recognized by the industry and has made considerable progress compared to manual compilation based on experience, there is still a lot of room for improvement in the transmission effect [4].

In recent years, related research has made some progress in solving dynamic-task-scheduling problems.

Zhou et al. summarize the application of an immune-optimization algorithm in unmanned aerial vehicles (UAVs)' scheduling problem. In the optimization process, the immune algorithm introduces an affinity-evaluation operator, an individual-concentration-evaluation operator and an incentive-evaluation operator before searching for the global-optimal solution using the mechanisms of population-diversity maintenance and parallel-distributed search [5]. Nazarov et al. use queuing theory to deal with the access-task-scheduling problems of database node services [6]. Liu et al. combine the theories of individual concentration and individual incentive degree in the immune algorithm with the fitness function in the genetic algorithm (GA) to both guide the algorithm's search process and achieve multi-objective task scheduling optimization [7]. Liu et al. use the evolutionary algorithm (EA) as a search engine for large-scale global optimization (LSGO) technology to find the global-optimal solution in complex high-dimensional spaces [8]. Jia et al. propose distributed cooperative co-evolution algorithms (DCC) to solve the optimization-evaluation problem, evaluate the contribution of each subgroup to the global fitness, and allocate the computing resources of the subpopulation according to their degree of contribution [9]. Sun et al. propose a threshold-based grouping strategy and grouping variables by pre-setting the relevant threshold of subgroups [10]. Li et al. solve the overlapping LSGO problem by setting a threshold to specify the size of subgroups. The decision variables are grouped by clustering to avoid the problem of uneven groupings [11]. In solving the multi-objective problems (MOP), the multi-objective evolutionary algorithm (MOEA) is adopted and the algorithm is improved, providing many ideas when studying the dynamic-task-scheduling problem [12–19]. Kuppusamy et al. introduce a reinforced strategy-dynamic-opposition learning based on social-spider-optimization algorithms to enhance individual superiority and schedule workflow in fog computing [20]. Tang et al. propose a job-scheduling algorithm based on the workload prediction of computing nodes, analyze the causes of workload imbalance and the feasibility of reallocating computing resources, and design an application and a workload-aware scheduling algorithm (AWAS) by combining the previously designed workload prediction models, and propose a parallel-job scheduling method based on computing-node-workload predictions [21]. Jia et al. study single-objective flexible job shop scheduling problems (JSP) by combining genetic algorithms and whale-swarm algorithms; reorganizing whale individuals and genetic codes to improve the local-search ability; and making comparisons with standard examples. Based on this, they conclude that the optimal solution can be obtained [22].

In the above research using EAs, GAs and other intelligent algorithms to solve the task-dynamic-scheduling problem, we found that the DSTT is most similar to the JSP. The algorithm for solving JSP problems has reference significance for the DSTT.

The classical job-shop-scheduling problem (JSP) is one of the most well-known scheduling problems. It consists of m machines and n jobs. A job contains several operations to be processed in a fixed order. In the JSP, each operation can be processed by one specific machine [23,24]. The flexible job shop scheduling problem (FJSP) is an extension of the JSP, which allows an operation to be processed by one of two or more machines. In other words, an operation is processed by one of the alternative machines in the FJSP [25]. Since a machine is predetermined for a specific operation, the JSP can be solved by specifying the priority of given operations such that a high-priority operation precedes others with a lower priority in the queue for given machines. Thus, the FJSP can be regarded as a sequencing problem, suitable for intelligent algorithms [26,27]. The genetic algorithm based on candidate sequence (COGA) is adopted to solve the FJSP [28].

Developed by Eberhart and Kennedy in 1995, particle-swarm optimization (PSO) is a population-based stochastic optimization technique inspired by the swarming behavior characteristic of bird flocks or fish schools [29]. PSO has recently been applied to COPs, such as shop scheduling [30,31], the traveling-salesman problem [32], quality-of-service multicast routing [33], and vehicle routing [34,35].

PSO is usually used to solve continuous problems, and discretization is required when solving COPs. Zheng et al. use PSO to solve the assembly JSP, design a discrete particle swarm optimization algorithm (DPSO) to realize the discretization of the position-update process through operations such as insertion and exchange. The Metropolis criterion in the simulated annealing algorithm (SA) is used to set the acceptance probability of the location update [36]. Chen et al. used GAs and DPSOs to manage the complexity of the problem, compute feasible and quasi-optimal trajectories for mobile sensors, and determine the demand for movement among nodes [37]. Fan et al. used DPSOs combined with the genetic operators of GAs to compute feasible and quasi-optimal schedules for directional sensors and to determine the sensing orientations among the directional sensors [38]. The above studies use other intelligent algorithms such as the SA and GA to hybrid with PSO, replacing particle updates with intelligent-algorithm operations. We need to make targeted improvements to the DPSO for specific problems.

1.2. Contributions

The following problems need to be solved when applying a DPSO to the DSTT:

1. It is necessary to establish a mathematical model suitable for the DPSO of the DSTT;
2. It is necessary to solve the DPSO's mathematical-description problems of particle position, direction, and velocity.
3. It is necessary to design specific particle-update methods.

Based on the above problems, this paper conducts research on the DSTT, and improves the DPSO. The contributions of this paper are mainly reflected in the following aspects:

1. We build a mathematical model of the DSTT, including a task model, an evaluation model, an evaluation function, and an output of the Schedule—the mathematical model of the transmission plan—and propose a one-dimensional code to describe the Schedule, making it suitable for the calculation of the DPSO;
2. We propose a DPSO to define the particle position, particle-update direction and target definition, and solve the mathematical description of the DPSO for specific problems;
3. Based on the basic idea of the DPSO, and taking into account inertia retention, particle best, and global best, we use the probability-selection model to realize the particle update, and we propose random perturbation to improve the diversity of the particle population.

In Section 2, we build a mathematical model of the DSTT suitable for intelligent-algorithm processing. In Section 3, we outline the specific operation and implementation method of the DPSO used in this paper and test the DPSO's optimal parameter combination. In Section 4, the experimental results are presented, and the results are analyzed and discussed. In Section 5, we make a conclusion and point out future avenues of work.

2. Preliminaries

The task-scheduling plan of a radio- and television-transmission station for daily execution of the transmission task is called the Schedule. According to the Chinese radio and television industry standard [39,40], the operation diagram is defined as follows:

Definition 1. *Schedule: a table that specifies the broadcasting tasks undertaken by the transmitter in a day according to the sequence of program broadcasting time.*

The mathematical model of the Schedule is a two-dimensional table; the horizontal axis represents the time, and the vertical axis represents the equipment code. The DSTT model can be described as the problem of filling the transmission task into the Schedule and maximizing the comprehensive-evaluation function. The DSTT model framework is shown in Figure 1.

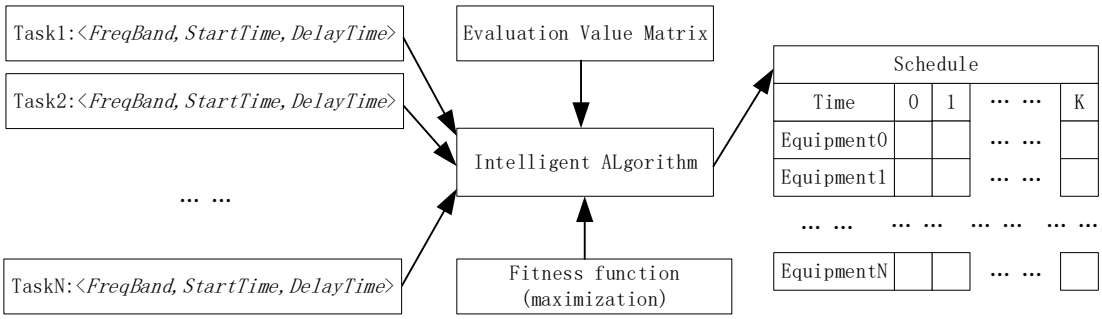

Figure 1. Model framework of DSTT.

The transmission-task sequence and the transmission-effect evaluation value matrix are the input data, and the maximum value of the evaluation function is taken as the fitness. Using intelligent-algorithm calculations, a Schedule meeting the requirements is obtained.

2.1. Task Queue

The task queue is the set of tasks to be executed every day. The parameters of the task model include the code of the working frequency band, the code of the task's start time, and the delay time of the task, which can be expressed in triples:

$$Task = < FreqBand, StartTime, DelayTime > \quad (1)$$

The task queue can be described as:

$$TaskQueue = \{Task_1, Task_2, \ldots, Task_j, \ldots Task_p\} \quad (2)$$

where, p is the number of tasks, $Task_j$ is the j-th of tasks, $Task_j.FreqBand$ is the frequency-band code of the working frequency of the task, $Task_j.StartTime$ is the code of the start time of the task, and $Task_j.DelayTime$ is the delay time of the task.

2.2. Value Matrix

In order to evaluate the transmission effect of the equipment in each operating frequency band, the value matrix is set, which is obtained by polling the transmission fre-

quency band during the trial operation of the station's transmission equipment. The value matrix can be described as follows:

$$ValueMatrix = \begin{bmatrix} Va_{11} & Va_{12} & \cdots & Va_{1i} & \cdots & Va_{1m} \\ Va_{21} & Va_{22} & \cdots & Va_{2i} & \cdots & Va_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ Va_{i1} & Va_{i2} & \cdots & Va_{ij} & \cdots & Va_{im} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ Va_{n1} & Va_{n2} & \cdots & Va_{nj} & \cdots & Va_{nm} \end{bmatrix} \quad (3)$$

where n is the number of equipment, m is the number of frequency-band divisions, and Va_{ij} is the evaluation value of the transmitting effect of the i-th equipment working in the j-th frequency band.

Based on the supervised-learning theory, the value of the matrix is weighted and adjusted in combination with the results collected from each daily transmission to ensure that the matrix is dynamically updated to meet the practical needs of the system.

2.3. Task-Scheduling Result

In the model design, it is assumed that the transmission tasks completed by all the equipment are fully loaded, that is, the design can cover all the equipment according to the length of the task's start and end. In order to adapt to the definition of population in intelligent algorithms and the requirements of intelligent-algorithm-related operations, a one-dimensional sequence is used to represent the optimization-result data. Define the scheduling-result data as binary:

$$TaskResult = < Task, TransNo > \quad (4)$$

where, $TaskResult.Task$ is the task of $TaskQueue$, $TaskResult.TransNo$ is the equipment code that carries the task, and the output-result sequence of the algorithm can be expressed as:

$$ScheduleResult = \{TaskResult_1.TransNo, \ldots, TaskResult_k.TransNo, \ldots TaskResult_p.TransNo\} \quad (5)$$

where, p is the number of tasks, $TaskResult_k$ represents the scheduling result of the k-th task, and $TaskResult_k.TransNo$ is the code of the equipment that carries the k-th task of the $TaskQueue$.

2.4. Schedule

Define the *Schedule* as a two-dimensional table:

$$Schedule = \begin{bmatrix} FreqBand_{11} & FreqBand_{12} & \cdots & FreqBand_{1i} & \cdots & FreqBand_{1m} \\ FreqBand_{21} & FreqBand_{22} & \cdots & FreqBand_{2i} & \cdots & FreqBand_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ FreqBand_{i1} & FreqBand_{i2} & \cdots & FreqBand_{ij} & \cdots & FreqBand_{im} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ FreqBand_{n1} & FreqBand_{n2} & \cdots & FreqBand_{nj} & \cdots & FreqBand_{nm} \end{bmatrix} \quad (6)$$

where, n is the number of equipment,m is the number of Schedule time divisions, and $FreqBand_{ij}$ is the transmission-frequency-band code of the i-th equipment in the j-th operation time. The value-taking formula is:

$$\begin{aligned} FreqBand_{ij} = \quad & TaskResult_k.Task.FreqBandk \in [1,\ p], i = TaskResult_k.TransNo, \\ & j \in [TaskResult_k.Task.StartTime, TaskResult_k.Task.StartTime \\ & + TaskResult_k.Task.DelayTime) \end{aligned} \quad (7)$$

where k represents the task-scheduling-result code from 1 to p, and $TaskResult_k.Task.FreqBand$ is the frequency-band code of the *Schedule* task represented by the result. The value of the

abscissa i of the *Schedule* is the equipment code assigned to the result, and the value of the ordinate j of the *Schedule* is the start time of the *Schedule* task represented by the result and is marked until the end of the task.

2.5. Fitness Function

Set the fitness function of the *Schedule* as the maximum value of the transmission-effect evaluation, described as follows:

$$Max : Value(ScheduleResult) = \frac{\sum_{k=1}^{p} Va_{TaskResult_k.TransNoTaskResult_k.Task.FreqBand} \times TaskResult_k.Task.DelayTime}{m \times n} \quad (8)$$

The subscripts of $Va_{TaskResult_k.TransNoTaskResult_k.Task.FreqBand}$ are $TaskResult_k.TransNo$, and $TaskResult_k.Task.FreqBand$. $TaskResult_k.TransNo$ represents the equipment code represented by the k-th value of the result sequence, $TaskResult_kTask.FreqBand$ represents the frequency-band number of the Schedule task represented by the k-th value of the result sequence, and the $Value(ScheduleResult)$ represents the comprehensive evaluation value of the Schedule.

According to the above model framework and the data model description, the data-structure parameters are outlined in Table 1.

Table 1. Parameter table of DSTT mathematical model.

Parameter	Explain
m	number of *Schedule* definition periods
n	number of equipment defined in *Schedule*
p	number of *Schedule* tasks
$Task_k$	task of the k-th Schedule, including $< TaFr, Start, Span >$
$TaskResult_k$	data description of the k-th result, including $< Task, TransNo >$
Va_{ij}	evaluation value of the transmitting effect of the i-th equipment in the j-th frequency band
$FreqBand_{ij}$	*Schedule* node data, frequency code of the i-th equipment working in the j-th time period
$ScheduleResult$	The output result of *Schedule* is composed of p piece of $TransNo$ code values

2.6. Example

Combined with the description of the mathematical model, an example of a DSTT is shown in Figure 2.

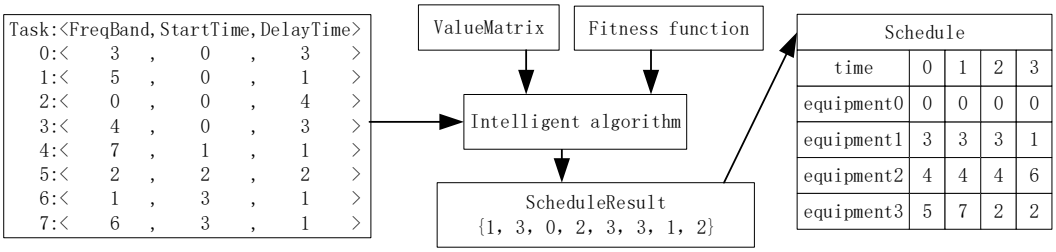

Figure 2. Example of DSTT.

According to the value matrix, under the condition of obtaining the maximum value for the fitness function of the Schedule, the task-dynamic-scheduling result is obtained through the intelligent algorithm. The result data are the equipment-code sequence. The two-dimensional table of the Schedule is shown in Figure 2. The abscissa is the sequence code of the period, the ordinate is the sequence code of the equipment, and the data are the frequency band transmitted by the equipment during the period. For example, the number of position 0 in the result value is 1, indicating that task0 is assigned to equipement1. The frequency-band code of task0 is 3, the start time is 0, and the delay time is 3. With time from 0 to 2 in equipment1 in the *Schedule*, the frequency-band code is 3.

3. Methodologies

3.1. PSO

PSO is a group-search-optimization algorithm. The motion of each particle is determined by the value of the fitness function, and the "direction" and "target" of its motion are determined by the "velocity" of each particle. Then, the particles iterate in the solution space according to the direction of the best particle.

In PSO, x represents the position of the particles, v represents the velocity of the particles, and *Pbest* represents the best position of the particles. The PSO initializes a group of random particles and finds the best solution through iteration. In each iteration, the particle updates its position by tracking two best values. One best value is the best solution that the particle can find. This solution is called particle best. The other best value is the best solution found by the whole population at present, which is called the global best. Suppose that a population composed of K particles is searched in the D-dimensional solution space, where the position of the i-th particle is expressed as a D-dimensional vector:

$$X_i = (x_{i1}, x_{i2}, \ldots, x_{iD}), i = 1, 2, \ldots, K \tag{9}$$

The motion velocity of the i-th particle is also a vector of the D-dimension:

$$V_i = (v_{i1}, v_{i2}, \ldots, v_{iD}), i = 1, 2, \ldots, K \tag{10}$$

The best position searched by the i-th particle, namely, the particle best, is expressed as:

$$Pbest_i = (p_{i1}, p_{i2}, \ldots, p_{iD}), i = 1, 2, \ldots, K \tag{11}$$

The best position searched by the whole population, namely, the global best, is expressed as:

$$Gbest = (g_1, g_2, \ldots, g_D) \tag{12}$$

The updated formula of velocity and position is as follows:

$$v_{id}^{t+1} = \omega * v_{id}^{t} + c_1 r_1 \left(p_{id}^{t} - x_{id}^{t}\right) + c_2 r_2 \left(g_d^{t} - x_{id}^{t}\right), \quad d = 1, 2, \ldots, D \tag{13}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t}, \quad d = 1, 2, \ldots, D \tag{14}$$

where c_1 and c_2 are the acceleration constant, r_1 and r_2 are a uniform random number, and ω is the inertia constant.

According to the above description, PSO is applicable to the continuous-function calculation, and the update of velocity and position adopts a continuous-vector calculation. Based on the discrete data characteristics of the DSTT, combined with the previous research, we carry out a targeted operation of the DPSO.

3.2. Algorithm Description

1. Definition of particle position: in combination with the DSTT and the example, see Formula (15) for the definition of particle position.

$$X_i = \left(x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{in}\right), i = 1, 2, \ldots, K,\ x_{ij} = Task_j.TransNo \quad n = TaskNumber \tag{15}$$

where K represents the population size, $TaskNumber$ represents the number of tasks, and $Task_j.TransNo$ represents the equipment code assigned to the j-th task. In Figure 2, the particle position is the allocation result of 8 tasks, and each number represents the equipment code assigned to the task. The queue is {1,3,0,2,3,3,1,2}, and each number is the equipment code assigned to each task.

The definitions of particle best and global best are the same as the definition of particle position, which is recorded as:

$$Pbest_i = \left(p_{i1}, p_{i2}, \ldots, p_{ij}, \ldots, p_{in}\right), i = 1, 2, \ldots, K \quad p_{ij} = Task_j.TransNo \tag{16}$$

$$Gbest = \left(g_1, g_2, \ldots, g_j, \ldots, g_n\right), \quad g_i = Task_j.TransNo \tag{17}$$

2. Definition of particle-motion direction: in combination with the characteristics of data discretization in DSTT, there is no direct association management between each task. Binary processing is adopted when defining particle-motion direction, that is, each particle-motion direction is each task that can change the equipment, and is recorded as:

$$VD_i = \left(vd_{i1}, vd_{i2}, \ldots, vd_{ij}, \ldots, vd_{in}\right), i = 1, 2, \ldots, K \quad vd_{ij} = 0, 1 \tag{18}$$

Only one of the vd_{ij}'s sequence values represented by each VD_i is 1, and the other is 0. Position 1 represents the task of replacing the node when the particle in *Schedule* updates the position, that is, the motion direction of the particle in *Schedule*. For example, $VD_i = (0, 0, 0, 1, 0, 0, 0, 0)$ indicates that the i-th particle in the population needs to replace the equipment working on Task2.

3. Definition of particle-motion target: the velocity displacement is defined as the serial number of the equipment to be replaced by the node representing the motion direction of particles in *Schedule*. The particle-motion target suitable for the operation of DSTT is defined as VT_i, which is recorded as:

$$VT_i = \left(vt_{i1}, vt_{i2}, \ldots, vt_{ij}, \ldots, vt_{in}\right), i = 1, 2, \ldots, K, vt_{ij} = -1, TransNo \tag{19}$$

Only one value of each VT_i represented by sequence vt_{ij} is recorded as *TransNo*, and the others are recorded as -1. The value taken by *TransNo* indicates that the node is replaced by the representative emission equipment when the particle in the *Schedule* updates its position, that is, the replacement target encoded by the particle's position in the particle's motion direction is defined as the particle-motion target in the *Schedule*. For example, $VT_i = (-1, -1, -1, 3, -1, -1, -1, -1)$ indicates that the i-th particle in the population wants to replace the equipment2 working on task3 with equipment3. An example of particle motion is shown in Figure 3.

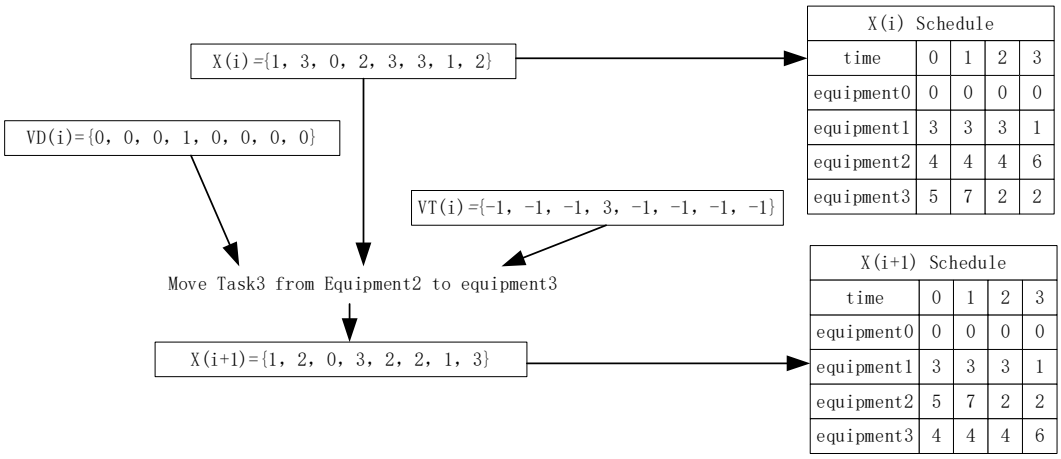

Figure 3. Example of particle motion.

Combining the problem characteristics of DSTT, particle motion has only one motion direction and target in order to ensure that the particle update of the DPSO have the characteristics of inertia preservation, particle best, and global best direction vector calculations included in the basic PSO. The DPSO uses a probabilistic-selection model to handle particle

updates, and determines the proportion of parameters by which particles choose one of the three directions to perform motion operations. If the result of the probabilistic-selection model is inertia retention or the particle-motion target is consistent with the current position, random-perturbation processing is introduced to increase the population particle's diversity and avoid entering the local-optimal trap.

4. Definition of particle-position update: according to the characteristics of DSTT, the operation of particle-velocity update is to calculate the motion direction of particles and the motion target of particles. The evaluation value of each particle task is calculated according to the particle position, and XV_i is recorded as:

$$XV_i = (xv_{i1}, xv_{i2}, \ldots, xv_{ij}, \ldots, xv_{in}), i = 1, 2, \ldots, K \quad xv_{ij} = Va_{x_{ij}Task_j.Freqband} \tag{20}$$

where x_{ij} is the equipment code currently assigned to the j-th task represented by the i-th particle in the particle-position definition, $Va_{x_{ij}Task_j.FreqBand}$ represents the evaluation value in *ValueMatrix* when the j-th task is executed by the x_{ij} equipment, and $Task_j.Freqband$ represents the frequency-band code of the task. Similarly, the evaluation-value sequence of particle best and global best is recorded as $PbestV_i$, $GbestV$:

$$PbestV_i = \left(pv_{i1}, pv_{i2}, \ldots, pv_{ij}, \ldots, pv_{in}\right), i = 1, 2, \ldots, K \quad pv_{ij} = Va_{p_{ij}task_j.FreqBand} \tag{21}$$

$$GbestV = (gv_1, gv_2, \ldots, gv_i, \ldots, gv_n), \quad gv_i = Va_{g_iTask_j.FreqBand} \tag{22}$$

where p_{ij} is the equipment code assigned to the j-th task indicated in the definition of the particle-best position of the i-th particle. The $Va_{p_{ij}Task_j.FreqBand}$ indicates the evaluation value in *ValueMatrix* when the j-th task is executed on the p_{ij}-th equipment, the $Task_j.FreqBand$ indicates the frequency-band code of the task. The definition of global best is consistent with that of particle best.

Calculate the evaluation-value gap between the particle-position and the particle-best values based on the difference between the above two evaluation value data series, and calculate the evaluation-value gap between the current particle and the global-best value based on the difference between the two data series. Select the maximum difference in evaluation values as the motion-direction option of particles.

Whether the particle moves towards the particle best direction or the global best direction depends on the output of the probability-selection model. The equipment code of particle best and global best directions is used as the motion target of particles. For example, the output of the probability-selection model is the global-best direction. Compare and calculate XV_i and $GbestV$. When $j = L$, $Va_{g_iTask_j.FreqBand} - Va_{x_{ij}Task_j.FreqBand}$ is at maximum value, then L is the motion direction of particles. The motion direction is $VD_i = (0, 0, \ldots, 1, \ldots, 0)$ where only the L-th value is 1, and the other value is 0. The motion target is $VT_i = (-1, -1, \ldots, Task_L.TransNo, \ldots, -1)$ where only the L-th value is $Task_L.TransNo$ of $Gbest$, and the other value is -1. Combined with DSTT, $Task_L.TransNo$ is the equipment code assigned to the L-th task with the largest difference between the particle position and the global best.

In DSTT, the goal of fitness function is the maximum evaluation value. The D-value between the evaluation values calculated before and after particle iteration can be understood as the motion distance of the particle.

3.3. Parameter

The DPSO is controlled by three parameters: inertia-retention factor (IRF), particle-best factor (PBF), and global-best factor (GBF). The three parameters correspond to ω, c_1r_1, and c_2r_2 in Formula (13). Set the sum of the three parameters to 1. Each iteration has a proportion of ω to perform inertia retention, a proportion of c_1r_1 to perform local-best motion, and a proportion of c_2r_2 to perform global-best motion. In the parameter experiment, we set the change-step size of the parameter to 0.1. The experiment sets

the number of equipment and the number of time periods to be equal. The enumeration algorithm is used to verify the interval between the number of time periods and the number of equipment [4, 7]. The enumeration algorithm cannot be tested due to the computing time periods including more than 8 pieces of equipment. Each task number randomly generates 100 task sequences for testing. The experimental data are shown in Table 2:

Table 2. Comparison of the number of successful tests of DPSO parameters.

Parameter			Number of Equipment and Time Periods				Total
IRF	PBF	GBF	4	5	6	7	
0.8	0.1	0.1	100	100	100	99	399
0.7	0.1	0.2	100	100	100	100	400
0.6	0.2	0.2	100	100	100	99	399
0.6	0.1	0.3	100	100	100	100	400
0.6	0.2	0.2	100	100	100	99	399
0.6	0.3	0.1	100	100	99	97	396
0.5	0.1	0.4	100	100	100	100	400
0.5	0.2	0.3	100	100	100	100	400
0.5	0.3	0.2	100	100	98	98	396
0.5	0.4	0.1	100	100	97	93	390
0.4	0.1	0.5	100	100	100	100	400
0.4	0.2	0.4	100	100	100	99	399
0.4	0.3	0.3	100	100	100	99	399
0.4	0.4	0.2	100	99	96	90	385
0.4	0.5	0.1	100	99	96	96	391
0.3	0.1	0.6	100	100	100	99	399
0.3	0.2	0.5	100	100	100	97	397
0.3	0.3	0.4	100	99	99	97	395
0.3	0.4	0.3	100	100	97	96	393
0.3	0.5	0.2	100	99	98	94	391
0.3	0.6	0.1	100	99	97	93	389
0.2	0.1	0.7	100	100	100	100	400
0.2	0.2	0.6	100	100	100	100	400
0.2	0.3	0.5	100	99	99	95	393
0.2	0.4	0.4	100	98	98	97	393
0.2	0.5	0.3	100	99	94	92	385
0.2	0.6	0.2	100	95	97	89	381
0.2	0.7	0.1	99	98	89	87	373
0.1	0.1	0.8	100	100	100	99	399
0.1	0.2	0.7	100	100	100	99	399
0.1	0.3	0.6	100	100	98	99	397
0.1	0.4	0.5	100	96	98	94	388
0.1	0.5	0.4	100	96	99	90	385
0.1	0.6	0.3	100	97	95	93	385
0.1	0.7	0.2	99	98	90	89	376
0.1	0.8	0.1	99	96	89	87	371

According to the number of equipment and time periods in the 4–7 interval, the enumeration algorithm is used to obtain the optimal values for the comparison tests. The iteration number of the parameter test table is obtained as follows. In order to better compare the change in the iteration number of different parameter groups, the iteration number is evaluated and calculated. The number of each equipment is calculated by dividing the iteration number by the average value of the iteration number obtained by all parameter groups. The cumulative-average proportion is shown in Table 3.

Table 3. Comparison table of iterations of DPSO.

Parameter			Number of Equipment and Time Periods				Average Weighted
IRF	**PBF**	**GBF**	**4**	**5**	**6**	**7**	
0.8	0.1	0.1	224	871	2655	21,824	3.9881
0.7	0.1	0.2	112	599	2217	10,548	2.3223
0.6	0.2	0.2	172	557	1650	8722	2.3816
0.6	0.1	0.3	121	591	1836	5129	1.9570
0.6	0.2	0.2	113	439	3923	9776	2.4696
0.6	0.3	0.1	178	452	3309	16,446	3.1171
0.5	0.1	0.4	149	512	1569	8560	2.1828
0.5	0.2	0.3	132	367	1469	5336	1.7257
0.5	0.3	0.2	118	432	2423	11,878	2.3078
0.5	0.4	0.1	112	307	6163	25,490	3.7425
0.4	0.1	0.5	122	522	1749	3791	1.7931
0.4	0.2	0.4	122	415	1270	7374	1.8020
0.4	0.3	0.3	124	334	975	8476	1.7363
0.4	0.4	0.2	119	780	8287	35,345	5.2982
0.4	0.5	0.1	109	1033	7520	21,687	4.5165
0.3	0.1	0.6	149	443	1619	7191	2.0399
0.3	0.2	0.5	110	453	1363	10,859	2.0050
0.3	0.3	0.4	127	843	3195	12,709	2.9862
0.3	0.4	0.3	104	351	6735	17,436	3.3739
0.3	0.5	0.2	754	628	4604	19,818	6.9327
0.3	0.6	0.1	103	1049	7460	22,095	4.5122
0.2	0.1	0.7	139	567	1450	4326	1.9032
0.2	0.2	0.6	109	415	1283	3175	1.4791
0.2	0.3	0.5	120	725	3347	14,749	2.9821
0.2	0.4	0.4	119	1037	4929	15,402	3.6608
0.2	0.5	0.3	109	554	9827	28,083	4.8906
0.2	0.6	0.2	107	4385	6990	37,551	8.7769
0.2	0.7	0.1	498	1632	16,425	46,448	10.6003
0.1	0.1	0.8	133	455	1978	5732	1.9499
0.1	0.2	0.7	115	378	1600	5762	1.6963
0.1	0.3	0.6	126	352	4289	4313	2.1971
0.1	0.4	0.5	146	2829	5560	13,826	5.6724
0.1	0.5	0.4	118	1822	2081	29,446	4.7190
0.1	0.6	0.3	116	3604	9047	19,192	7.3423
0.1	0.7	0.2	614	2266	15,314	39,792	11.2526
0.1	0.8	0.1	612	2278	18,466	36,460	11.7022

It can be seen from Table 3 that the number of iterations in the parameter groups' calculations to obtain all global-best solutions in Table 2 is also small, indicating that the algorithm's success rate and efficiency are unified within the same parameter group.

It is impossible to enumerate the parts of the algorithm's comparison experiments and to compare whether the evaluation values obtained under different parameters obtained the maximum value for all parameters. The number of iterations of the algorithm is the number of iterations of the greedy algorithm. The formula is as follows:

$$IterationNumber = \sum_{i=1}^{n} i^2 \tag{23}$$

The number of experimental tasks includes the data from the previous 4–7 intervals. According to the computing power of the current experimental environment, the number of equipment and the number of time periods during the experiment is 12. The data can be found in Table 4.

Table 4. Comparison of times required to obtain the best value.

Parameter			Number of Equipment and Time Periods									Total
IRF	**PBF**	**GBF**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	
0.8	0.1	0.1	√	√	√							3
0.7	0.1	0.2	√	√	√	√						4
0.6	0.2	0.2	√	√	√							3
0.6	0.1	0.3	√	√	√	√						4
0.6	0.2	0.2	√	√	√							3
0.6	0.3	0.1	√	√								2
0.5	0.1	0.4	√	√	√	√	√		√			6
0.5	0.2	0.3	√	√	√	√						4
0.5	0.3	0.2	√	√								2
0.5	0.4	0.1	√	√								2
0.4	0.1	0.5	√	√	√	√	√				√	6
0.4	0.2	0.4	√	√	√							3
0.4	0.3	0.3	√	√	√							3
0.4	0.4	0.2	√									1
0.4	0.5	0.1	√									1
0.3	0.1	0.6	√	√	√			√		√		5
0.3	0.2	0.5	√	√	√							3
0.3	0.3	0.4	√									1
0.3	0.4	0.3	√	√								2
0.3	0.5	0.2	√									1
0.3	0.6	0.1	√									1
0.2	0.1	0.7	√	√	√	√	√					5
0.2	0.2	0.6	√	√	√	√						4
0.2	0.3	0.5	√									1
0.2	0.4	0.4	√									1
0.2	0.5	0.3	√									1
0.2	0.6	0.2	√									1
0.2	0.7	0.1										0
0.1	0.1	0.8	√	√	√							3
0.1	0.2	0.7	√	√	√							3
0.1	0.3	0.6	√	√								2
0.1	0.4	0.5	√									1
0.1	0.5	0.4	√									1
0.1	0.6	0.3	√									1
0.1	0.7	0.2										0
0.1	0.8	0.1										0

Analysis of parameters' test results:

1. According to Table 2, when the number of equipment and time periods is [4, 7], the influence of the parameters on the algorithm results is small. When the PBF is small and the GBF is large, the algorithm's success rate is high. Taking the IRF of 0.1 as an example, as the PBF increases, the GBF decreases, and the success rate of the algorithm decreases gradually.
2. According to Table 3, when the number of equipment and time periods is small [4, 7], the algorithm performance is evaluated by the weighted average of the number of iterations of the algorithm required to reach the global-best solution. The algorithm parameters have a great impact on the number of iterations of the algorithm. Among the parameter groups with a success rate of 100%, the parameter groups (0.2,0.2,0.6) have the lowest number of iterations.
3. According to Table 4, the interval between the number of equipment and the number of time periods [4, 12] is considered globally. In the test of fixed iterations, the DPSO only achieves the maximum value of multiple parameters when the number of equipment and the number of time periods are 8. In [9, 12], some different parameter

groups achieved maximum values, including (0.3,0.1,0.6), (0.5,0.1,0.4), (0.2,0.1,0.7), and (0.4,0.1,0.5).

4. The groups (0.3,0.1,0.6) are better in the iterative-weighting calculation in the previous two comparison tables, but the success rate is slightly lower. The groups (0.2,0.2,0.6)' weighted-average number of iterations is minimal. In order to ensure the comprehensiveness of the subsequent multi-algorithm experiments, use 5 groups of parameters to carry out experiments in the subsequent comparison experiments of DPSO. The list is in Table 5.

Table 5. Comprehensive comparison table of parameter tests of DPSO.

	Parameter			[4, 7]		[4, 12]
	IRF	**PBF**	**GBF**	**Success Rate**	**Weighted Iteration**	**Maximum Number of Times**
DPSO1	0.5	0.1	0.4	100%	2.1828	6
DPSO2	0.4	0.1	0.5	100%	1.7931	6
DPSO3	0.3	0.1	0.6	99.75%	2.0399	5
DPSO4	0.2	0.1	0.7	100%	1.9032	5
DPSO5	0.2	0.2	0.6	100%	1.4791	4

4. Experiments

4.1. Comparison Algorithm and Method

The iteration number of each algorithm in this part is limited by Formula (23). The comparison algorithms can be found in Table 6.

Table 6. DSTT comparison algorithm list.

Algorithm	Explain
ENU	Enumeration algorithm: The global-optimal solution can be obtained by traversing the running chart of all tasks and calculating the evaluation value. With the increase in the number of equipment and time periods, the iteration number increases rapidly and the algorithm's execution time is long.
GR	Greedy algorithm: Find out the tasks that can obtain the best allocation among all tasks. According to this principle, until all tasks are allocated, the algorithm's execution time is stable, the number of algorithm iterations is related to the task period, and the conflicts encountered in the allocation are backtracked [4].
IGA	Improved genetic algorithm: Select the elitist-retention strategy, the discontinuous cycle replacement group crossover strategy for crossover, and the overall equipment task switching strategy for mutation. The three parameters are set as the algorithm's optimal parameter array, with a selection factor = 0.8, a crossover factor = 0.1, and a mutation factor = 0.1. Refer to previous research results for the selection of parameters [3].
DPSO	Discrete particle swarm optimization algorithm: The parameters are calculated according to the five sets of parameters in Table 5, and the corresponding statistical calculations are performed.

According to the characteristics of the DSTT, the algorithm-comparison test uses the enumeration algorithm as the reference algorithm in the range of [4, 7] between the number of equipment and the number of time periods. The enumeration algorithm takes a long time to produce its calculations, but it can clearly obtain the global-optimal solution. With the optimal solution as a comparison reference, the performance index of the algorithm is evaluated by recording the number of iterations required by the other algorithms to reach the optimal solution.

In the interval where the number of equipment and periods exceeds seven, there is no optimal solution for reference. The comparison method adopts two calculations. Each comparison is completed with each parameter group first. The comparison results show that the calculation results of more than two groups of parameters are completely consistent, and the result is determined to be the global best solution. Then, the second calculation is completed to obtain the number of iterations of the algorithm for each parameter to obtain the global-best solution so as to evaluate the algorithm's performance.

In order not to lose fairness, all algorithms in the test experiment in this section use the same initialization task queue for their calculations. Each task number randomly generates 100 task sequences for the best-scheduling calculations; compares the success rate of each algorithm in calculating the global-best solution under the above conditions; calculates the iteration times of the algorithm in the global-best solution; and calculates the proportional cumulative evaluation value, which represents the average of the calculated comprehensive evaluation value of the best allocation. Because the global best solution comparison is adopted for the whole interval, the interval is no longer distinguished, and the data are uniformly compared for the whole interval.

4.2. Algorithm Comparison Experiment

This section compares the effectiveness, accuracy, and efficiency of several algorithms in the experiments.

The number of successes in obtaining the global best solution from the experimental calculations are shown in Table 7.

Table 7. Multi-algorithm success number comparison table.

Eq.	ENU	GR	IGA	DPSO		
				[min, max]	Avg.	Stdea.
4	100	25	100	[100, 100]	100	0.000000
5	100	8	100	[100, 100]	100	0.000000
6	100	5	100	[99, 100]	99.8	0.447214
7	100	2	99	[98, 100]	99.6	0.894427
8	N/A	1	89	[99, 100]	99.8	0.447214
9	N/A	0	70	[92, 99]	97.2	2.949576
10	N/A	0	62	[93, 100]	97.8	2.774887
11	N/A	0	35	[93, 98]	96.8	2.167948
12	N/A	0	39	[83, 97]	93.2	5.932959
13	N/A	0	20	[90, 100]	94.8	3.701351
14	N/A	0	4	[77, 97]	88.6	8.049845
15	N/A	0	10	[67, 94]	87.4	11.436783
Total	400	41	728	[1124, 1182]	1155	26.870058

When the number of equipment is greater than 7, the ENU is no calculated due to too long time. The DPSO uses the five sets of parameters in Table 5 to calculate the success numbers and performs statistical calculations on the maximum, minimum, average, and mean square deviation of the five sets of result data. It can be seen that the DPSO's algorithm data are stable, but data stability decrease as the number of equipment increases. Comparing algorithms, the DPSO is superior to the IGA, and the calculation results of the two intelligent algorithms are far superior to that of the GR.

The average comparison of the evaluation values calculated by the algorithms is shown in Table 8.

Table 8. Comparison table of evaluation values of multi-algorithms.

Eq.	GR	IGA	DPSO		
			[min, max]	Avg.	Stdea.
4	0.815214	0.838134	[0.838134, 0.838134]	0.838134	0.000000
5	0.874183	0.895642	[0.895642, 0.895642]	0.895642	0.000000
6	0.856763	0.883459	[0.883458, 0.883459]	0.8834588	0.000000
7	0.899497	0.916141	[0.916114, 0.916156]	0.9161528	0.000007
8	0.892468	0.913266	[0.91355, 0.913556]	0.9135548	0.000003
9	0.858319	0.881545	[0.882189, 0.882289]	0.8822652	0.000043
10	0.874637	0.898784	[0.89988, 0.899956]	0.8999352	0.000031

Table 8. *Cont.*

Eq.	GR	IGA	DPSO		
			[min, max]	Avg.	Stdea.
11	0.873344	0.896935	[0.898354, 0.898388]	0.898379	0.000014
12	0.878253	0.902085	[0.903259, 0.90338]	0.9033514	0.000052
13	0.878391	0.914041	[0.916594, 0.916635]	0.9166176	0.000018
14	0.904408	0.931707	[0.934612, 0.934725]	0.9346716	0.000048
15	0.891888	0.917394	[0.920402, 0.920621]	0.920569	0.000094
Avg.	0.874780	0.899094	[0.900205, 0.900242]	0.9002276	0.000018

According to the data in Table 8, the evaluation-value data calculated by the DPSO algorithm are stable, indicating that the algorithm has high stability. Compared to the IGA, when the number of equipment is small, the accuracy of the two intelligent algorithms is equivalent. As the number of equipment increases, the accuracy advantage of the DPSO significantly increases. The calculation results of the GR algorithm differ greatly from those of the two intelligent algorithms.

The comparison results of the number of iterations to obtain the best solution are shown in Table 9.

Table 9. Multi-algorithm iteration number comparison table.

Eq.	IGA	DPSO		
		[min, max]	Avg.	Stdea.
4	643	[105, 132]	116.4	10.0
5	2692	[340, 552]	446.8	88.4
6	7452	[1674, 2088]	1812.4	179.8
7	15,095	[4355, 9186]	5893.8	1963.8
8	74,431	[7719, 11,447]	9395.8	1465.2
9	266,112	[36,169, 69,736]	48,312.8	13,134.6
10	421,878	[42,925, 88,540]	63,079	16,922.1
11	899,103	[108,956, 208,574]	132,676.2	42,597.5
12	1,210,932	[209,287, 407,190]	309,613.4	90,217.6
13	2,211,539	[362,240, 765,968]	498,789	166,816.1
14	3,241,904	[493,569, 1,381,226]	841,163.8	328,465.6
15	4,141,849	[735,817, 2,595,418]	1,297,816.8	746,327.9

This research uses the number of iterations to evaluate the efficiency of the algorithm, because intelligent algorithms need to calculate the results of the fitness function for each iteration, and generally the algorithm's operation time is much lower than the calculation time of the fitness function. Table 9 shows the statistical analysis of the iterations of the two intelligent algorithms. It can be seen that the iterations of the DPSO are far superior to the efficiency of the IGA. Because the DPSO is a cluster search and the number of iterations of the algorithm includes randomness, there is a large gap in the data statistics of DPSO algorithms.

4.3. Summary of Experimental Analysis

Within the range of the number of equipment and time periods that can be verified by the enumeration algorithm, and based on an analysis of success, the DPSO and the IGA are effective, and the approximate rates can calculate the global-optimal solution, while the GR is less effective. Within the range of the number of equipment and time periods that cannot be verified by the enumeration algorithm, the DPSO can calculate and obtain a global-optimal solution in most cases based on the disaster-recovery-backup idea and the analysis of the algorithm itself. As the number of equipment increases, the effectiveness of IGA algorithms significantly decreases.

Based on the analysis of the evaluation values, the evaluation values calculated by the DPSO and the IGA are close, while the GR's calculation results differ greatly from those calculated by the two intelligent algorithms. In all equipment-count tests, the DPSO achieved higher evaluation values than the IGA. This indicates that the accuracy of the DPSO is consistently higher than that of the IGA.

By analyzing the efficiency of the two intelligent algorithms through the number of iterations, the execution efficiency of the IGA differs greatly from that of the DPSO.

The advantages and disadvantages of various algorithms and comparison algorithms proposed in this paper are shown in Table 10.

Table 10. Comparison of advantages and disadvantages of multiple algorithms.

Eq.	Availability	Accuracy	Efficiency	Evaluation
ENU	high	N/A	N/A	bad
GR	low	low	N/A	bad
IGA	middle	high	low	better
DPSO	high	high	high	best

In comparative analyses, the DPSO algorithm is comprehensively evaluated to be the best.

4.4. Simulation Experiment

This section compares the advantages and disadvantages of several algorithms by simulating the actual situation of the transmitting station.

The experimental program writing tool used is Visual Studio 2012, and the language used is C++ with MFC architecture. The hardware environment used is a Microsoft Surface X1 portable computer, the CPU used is Intel Core i7 3.60 GHz, the memory used is 16 GB, and the operating system used is Win10.

In the actual working environment of the transmission station, the number of equipment is usually fixed, and the number of time periods varies according to the program settings. In this part of the simulation experiment, the parameters with the best performance of the algorithm are selected for the simulation experiment. The enumeration algorithm was abandoned due to the long execution time after the number of tasks increased, and the GR had no parameters. The IGA parameters are (0.8,0.1,0.1) for the simulation experiments, and the DPSO selected parameters (0.3,0.1,0.6) for the simulation experiments. The fixed value of the number of equipment is 10, and the number of time periods is 24 h per day, with one time period every half an hour, that is, a [4, 48] interval. The two intelligent algorithms are tested with the same number of iterations. Because there is no optimal-value comparison, the success-rate data cannot be compared. The simulation experiment only compares the algorithm's accuracy with the average value of the comprehensive evaluation value obtained 100 times. The data obtained are in Table 11.

Table 11. Accuracy comparison data table of three algorithms simulation experiments.

Periods	4	5	6	7	8	9	10	11	12
GR	0.893972	0.901704	0.913778	0.881862	0.879343	0.889435	0.906143	0.884089	0.896052
IGA	0.917416	0.92164	0.934022	0.90367	0.904581	0.907794	0.932187	0.911542	0.920281
DPSO	0.918031	0.922333	0.935052	0.904655	0.905095	0.909001	0.933586	0.912695	0.921244
Periods	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
GR	0.893406	0.881768	0.89654	0.907105	0.89874	0.860929	0.864331	0.894956	0.8873
IGA	0.921019	0.910778	0.916936	0.93331	0.920762	0.88272	0.890827	0.921293	0.914671
DPSO	0.922169	0.911839	0.91786	0.93453	0.921716	0.883748	0.891933	0.922291	0.915707

Table 11. *Cont.*

Periods	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
GR	0.885542	0.885767	0.885219	0.87728	0.880179	0.87955	0.852032	0.883815	0.863596
IGA	0.910287	0.912008	0.910358	0.904637	0.912072	0.902095	0.884231	0.909096	0.888886
DPSO	0.911239	0.913564	0.911396	0.906042	0.912925	0.90334	0.885226	0.910536	0.889882
Periods	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
GR	0.890629	0.89359	0.857689	0.866087	0.873645	0.883004	0.867741	0.88413	0.867158
IGA	0.915545	0.91956	0.885552	0.889054	0.899454	0.910442	0.890941	0.914558	0.897373
DPSO	0.916702	0.920654	0.886297	0.89005	0.900186	0.911575	0.891782	0.915296	0.898685
Periods	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**
GR	0.862168	0.867621	0.86952	0.848084	0.876947	0.863869	0.875981	0.905513	0.858482
IGA	0.884312	0.896044	0.895931	0.872597	0.907236	0.887478	0.899854	0.930317	0.889568
DPSO	0.884925	0.896671	0.897014	0.873367	0.908403	0.888566	0.900728	0.931161	0.890556

According to Table 11, the DPSO has obtained an evaluation value superior to that of the IGA, which shows that the two intelligent algorithms operate stably, and that the DPSO has obvious accuracy advantages when dealing with DSTTs. See Figure 4 for a comparison diagram.

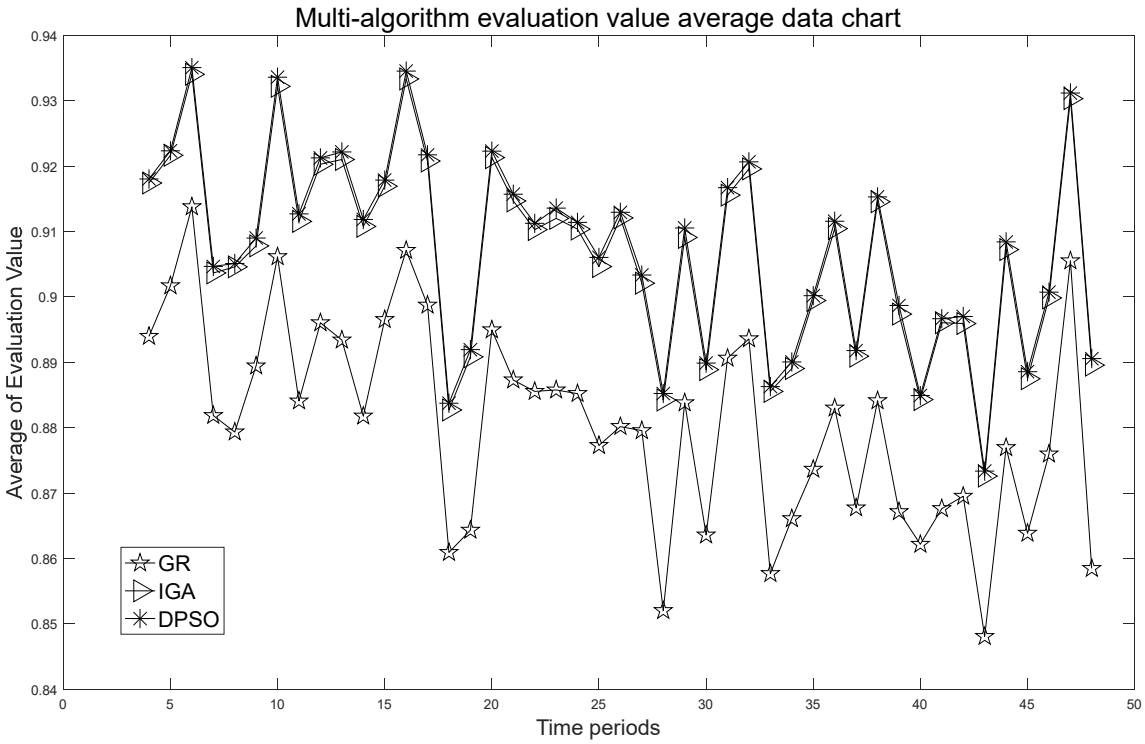

Figure 4. Multi-algorithm evaluation value comparison chart.

In the simulation experiments, select 100 experimental data with the largest number of time periods, that is, 48 time periods, and observe the evaluation value and algorithm calculation time of each of the two algorithms. The data are presented in Figure 5.

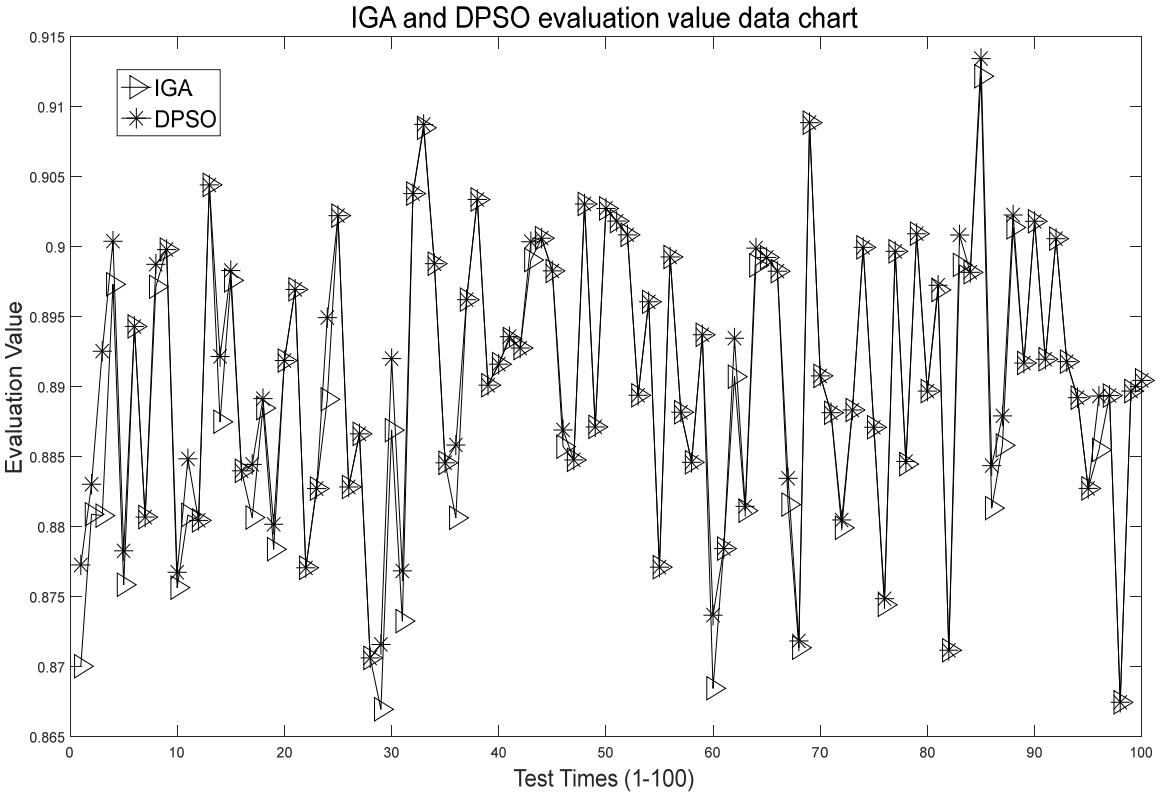

Figure 5. Two-algorithm evaluation-value comparison chart of 100 tests.

According to Figures 4–6, the following statistics can be obtained:

1. In the simulation experiment of multiple time periods, the evaluation values of the transmission effect for all time periods are averaged. The DPSO improved the evaluation value by 3.012295% compared to the GR, and the DPSO improved the evaluation value by 0.1111146% compared to the IGA.
2. In the simulation experiment with a maximum period of 48, the results of the DPSO were better than that of the IGA. In almost 60% of the 100 experiments, the IGA achieved the same results as the DPSO while other results were lower than that of the DPSO. The DPSO improved the evaluation value by 0.11115% compared to the IGA.
3. In the simulation experiment with a maximum period of 48, in 100 experiments the average execution time of the DPSO was 3.195 s, and the average execution time of the IGA was 10.388 s. The DPSO improved the execution efficiency by 69.246%.

According to the simulation experiment, it can be concluded that the intelligent algorithm and mathematical model proposed can solve the DSTT problem of radio- and television-transmission stations. The algorithm is stable in many simulation experiments, and the DPSO has the highest accuracy and the shortest execution time.

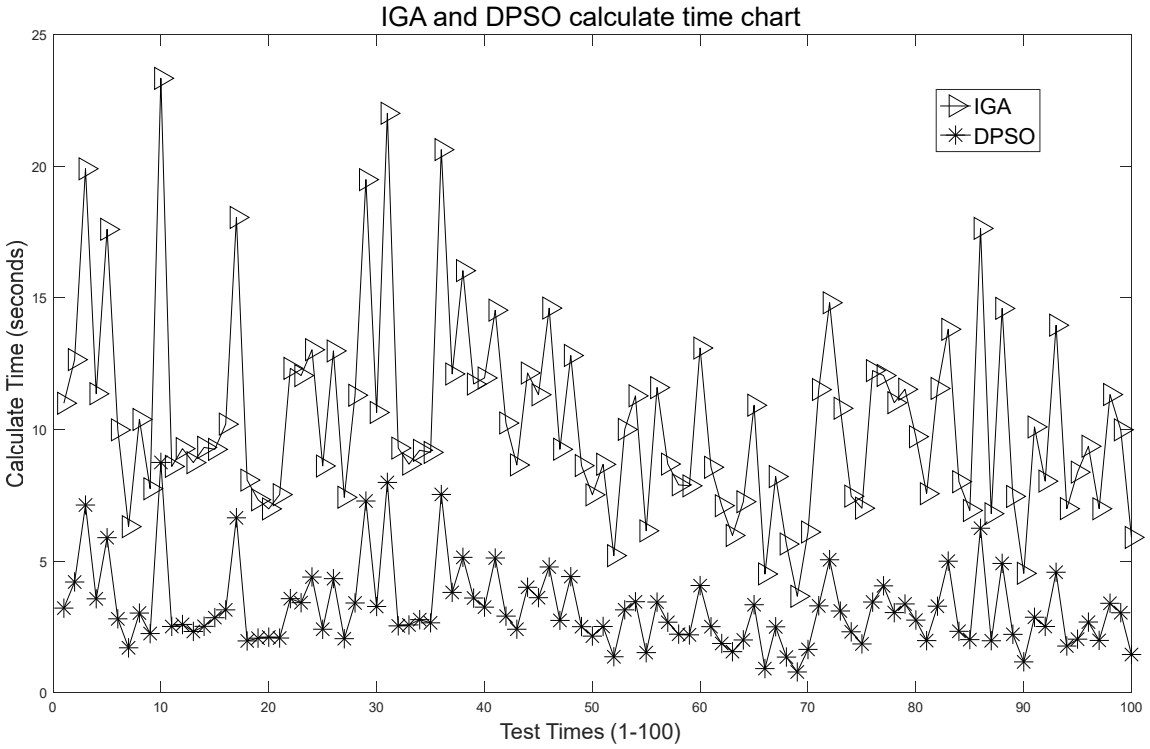

Figure 6. Two-algorithm calculate-time comparison chart of 100 tests.

5. Conclusions

The paper conducts research on the DSTT and designs a DSTT mathematical model suitable for operation by an intelligent algorithm. We proposed a fitness function with the goal of achieving the highest-effectiveness evaluation value. Based on PSO and recent research on the DPSO, the DPSO is specifically proposed for the DSTT.

The DPSO redefines particle-motion direction and particle-motion target based on the two-dimensional *Schedule*. Based on the characteristics of discrete problems, it proposes a probability-selection model to solve the specific operation of particle-position updates in discontinuous problems. It sets three parameters, namely, the inertia-retention factor, the particle-best factor and the global-best factor, to control the particle-position update to meet the idea of the DPSO, and adopts a random-perturbation operation. It avoids particle motion falling into the trap of local-optimal solutions. In parameter testing, the traversal method is used to determine the parameter group with the best calculation effect.

Finally, this paper outlines comparison experiments. In comparison experiments with GRs and IGAs, the effectiveness of the algorithm is verified using the success rate. Computational accuracy is verified using the calculation of evaluation values, and efficiency is compared using algorithm iterations. The results show that the DPSO has significant advantages in these three aspects.

The research results of this paper have certain reference significance for task scheduling and the COP in other industries.

The next research direction is to conduct multi-objective optimization research based on the research results of this paper, while ensuring the transmission effect and increasing the utilization rate of the equipment.

Author Contributions: Conceptualization, W.Y. and X.W.; methodology, W.Y. and X.W.; software, X.W.; validation, X.W.; formal analysis, W.Y.; investigation, W.Y. and X.W.; data curation, X.W.; writing—original draft preparation, X.W.; writing—review and editing, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Yang, M.; Xing, G.; Pang, J. Design and Implementation of Software Architecture of Intelligent Scheduling System for SW Broadcasting. *Radio TV Broadcast Eng.* **2007**, *4*, 112–116.
2. Hao, W. Application of Artificial Intelligence in Radio and Television Monitoring and Supervision. *Radio TV Broadcast Eng.* **2019**, *46*, 126–128.
3. Wang, X.; Yao, W. Research on Transmission Task Static Allocation Based on Intelligence Algorithm. *Appl. Sci.* **2023**, *13*, 4058. [CrossRef]
4. Zhou, D.; Song, J.; Lin, C.; Wang, X. Research on Transmission Selection Optimized Evaluation Algorithm of Multi-frequency Transmitter. In *2015 International Conference on Automation, Mechanical Control and Computational Engineering*; Atlantis Press: Amsterdam, The Netherlands, 2015; pp. 323–326.
5. Zhou, Z.; Luo, D.; Shao, J. Immune genetic algorithm based multi-UAV cooperative target search with event-triggered mechanism. *Phys. Commun.* **2020**, *41*, 101103. [CrossRef]
6. Nazarov, A.; Sztrik, J.; Kvach, A. Asymptotic analysis of finite-source M/M/1 retrial queueing system with collisions and server subject to breakdowns and repairs. *Ann. Oper. Res.* **2019**, *277*, 213–229. [CrossRef]
7. Zhao, X.; Xia, X.; Wang, L. A fuzzy multi-objective immune genetic algorithm for the strategic location planning problem. *Clust. Comput.* **2019**, *22*, 3621–3641. [CrossRef]
8. Liu, W.; Zhou, Y.; Li, B. Cooperative Co-evolution with soft grouping for large scale global optimization. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation, Wellington, New Zealand, 10–13 June 2019; pp. 318–325.
9. Jia, Y. Distributed cooperative co-evolution with adaptive computing resource allocation for large scale optimization. *IEEE Trans. Evol. Comput.* **2019**, *23*, 188–202. [CrossRef]
10. Sun, Y.; Li, X.; Ernst, A.; Omidvar, M.N. Decomposition for large-scale optimization problems with overlapping components. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation, Wellington, New Zealand, 10–13 June 2019; pp. 326–333.
11. Li, L.; Fang, W.; Wang, Q.; Sun, J. Differential grouping with spectral clustering for large scale global optimization. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation, Wellington, New Zealand, 10–13 June 2019; pp. 334–341.
12. Ismayilov, G.; Topcuoglu, H.R. Neural network-based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing. *Future Gener. Comput. Syst.* **2020**, *102*, 307–322. [CrossRef]
13. Cabrera, A.; Acosta, A.; Almieida, F. A dynamic multi-objective approach for dynamic load balancing in heterogeneous systems. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 2421–2434. [CrossRef]
14. Zhang, Q.; Yang, S.; Jiang, S. Novel prediction strategies for dynamic multi-objective optimization. *IEEE Trans. Evol. Comput.* **2019**, *24*, 260–274. [CrossRef]
15. Cao, L.; Xu, L.; Goodman, E.D. Evolutionary dynamic multi-objective optimization assisted by a support vector regression predictor. *IEEE Trans. Evol. Comput.* **2019**, *24*, 305–319. [CrossRef]
16. Qu, B.; Li, G.; Guo, Q. A niching multi-objective harmony search algorithm for multimodal multi-objective problems. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation, Wellington, New Zealand, 10–13 June 2019; pp. 1267–1274.
17. Liang, J.; Xu, W.; Yue, C. Multimodal multi-objective optimization with differential evolution. *Swarm Evol. Comput.* **2019**, *44*, 1028–1059. [CrossRef]
18. Qu, B.; Li, C.; Liang, J. A self-organized speciation-based multi-objective particle swarm optimizer for multimodal multi-objective problems. *Appl. Soft Comput.* **2020**, *86*, 105886. [CrossRef]
19. Ishibuchi, H.; Peng, Y.; Shang, K. A scalable multimodal multi-objective test problem. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation, Wellington, New Zealand, 10–13 June 2019; pp. 310–317.
20. Kuppusamy, P.; Kumari, N.M.J.; Alghamdi, W.Y.; Alyami, H.; Ramalingam, R.; Javed, A.R.; Rashid, M. Job scheduling problem in fog-cloud-based environment using reinforced social spider optimization. *J. Cloud Comput.* **2022**, *11*, 99. [CrossRef]
21. Tang, X.; Liu, Y.; Deng, T.; Zeng, Z.; Huang, H.; Wei, Q.; Li, X.; Yang, L. A job scheduling algorithm based on parallel workload prediction on computational grid. *J. Parallel Distrib. Comput.* **2023**, *171*, 88–97. [CrossRef]
22. Jia, P.; Wu, T. A hybrid genetic algorithm for flexible job-shop scheduling problem. *J. Xi'an Polytech. Univ.* **2020**, *10*, 80–86.
23. Cheng, R.; Gen, M.; Tsujimura, Y. A Tutorial Survey of Job-Shop Scheduling Problems using Genetic Algorithms-I. Representation. *Comput. Ind. Eng.* **1996**, *30*, 983–997. [CrossRef]
24. Cheng, R.; Gen, M.; Tsujimura, Y. A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: Hybrid genetic search strategies. *Comput. Ind. Eng.* **1999**, *36*, 343–364. [CrossRef]
25. Chaudhry, I.; Khan, A. A research survey: Review of flexible job shop scheduling techniques. *Int. Trans. Oper. Res.* **2016**, *23*, 551–591. [CrossRef]

26. Kim, J. Developing a job shop scheduling system through integration of graphic user interface and genetic algorithm. *Mul-timed. Tools Appl.* **2015**, *74*, 3329–3343. [CrossRef]
27. Kim, J. Candidate Order based Genetic Algorithm (COGA) for Constrained Sequencing Problems. *Int. J. Ind. Eng. Theory Appl. Pract.* **2016**, *23*, 1–12.
28. Park, J.; Ng, H.; Chua, T.; Ng, Y.; Kim, J. Unifified Genetic Algorithm Approach for Solving Flexible Job-Shop Scheduling Problem. *Appl. Sci.* **2021**, *11*, 6454. [CrossRef]
29. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
30. Pan, Q.; Tasgetiren, M.F.; Liang, Y.C. A discrete particle swarm optimization algorithm for the permutation flowshop sequencing problem with makespan criterion. In *Research and Development in Intelligent Systems XXIII, SGAI 2006*; Bramer, M., Coenen, F., Tuson, A., Eds.; Springer: London, UK, 2006; pp. 19–31.
31. Lian, Z.; Gi, X.; Jiao, B. A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan. *Chaos Solitons Fractals* **2008**, *35*, 851–861.
32. Shi, X.; Liang, Y.; Lee, H.; Lu, C.; Wang, Q. Particle swarm optimization-based algorithms for TSP and generalized TSP. *Inf. Process. Lett.* **2007**, *103*, 169–176. [CrossRef]
33. Abdel-Kader, R.F. Hybrid discrete PSO with GA operators for efficient QoS-multicast routing. *Ain Shams Eng. J.* **2011**, *2*, 21–31. [CrossRef]
34. Roberge, V.; Tarbouchi, M.; Labonte, G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans. Ind. Inf.* **2013**, *9*, 132–141. [CrossRef]
35. Xu, S.H.; Liu, J.P.; Zhang, F.H.; Wang, L.; Sun, L.J. A combination of genetic algorithm and particle swarm optimization for vehicle routing problem with time windows. *Sensors* **2015**, *15*, 21033–21053. [CrossRef]
36. Zheng, P.; Zhang, P.; Wang, M.; Zhang, J. A Data-Driven Robust Scheduling Method Integrating Particle Swarm Optimization Algorithm with Kernel-Based Estimation. *Appl. Sci.* **2021**, *11*, 5333. [CrossRef]
37. Chen, H.-W.; Liang, C.-K. Genetic Algorithm versus Discrete Particle Swarm Optimization Algorithm for Energy-Efficient Moving Object Coverage Using Mobile Sensors. *Appl. Sci.* **2022**, *12*, 3340. [CrossRef]
38. Fan, Y.-A.; Liang, C.-K. Hybrid Discrete Particle Swarm Optimization Algorithm with Genetic Operators for Target Coverage Problem in Directional Wireless Sensor Networks. *Appl. Sci.* **2022**, *12*, 8503. [CrossRef]
39. *GY/T 280-2014*; Specifications of Interface Data for Transmitting Station Operation Management System. State Administration of Press, Publication, Radio, Film and Television: Beijing, China, 2 November 2014.
40. *GY/T 290-2015*; Specification of Code for Data Communication Interface of Radio and Television Transmitter. State Administration of Press, Publication, Radio, Film and Television: Beijing, China, 3 March 2015.