Artificial Intelligence Scheduling Algorithms for Elevator Group Control Based on Finite Element Dynamic Analysis

Zhenlai Liu [1*]

[1]Department of electrical and mechanical engineering, Chongqing Aerospace Polytechnic, Chongqing, 400030, China

[*]fanxuechun139@163.com

[*]corresponding author

Keywords: Finite Element Dynamic Analysis; Elevator Group Control; Artificial Intelligence Scheduling Algorithm

Abstract: As a necessary equipment in modern high-rise buildings, elevators provide convenience for people and increase energy consumption. Therefore, how to properly dispatch elevators and achieve optimal group control has become one of the main topics in the construction industry. In this context, the problem of poor scheduling performance of traditional scheduling algorithms is caused, resulting in poor elevator group control performance. An elevator group control artificial intelligence scheduling algorithm based on finite element dynamic analysis is proposed. The algorithm firstly analyzes the finite element dynamics of elevator group control, and then uses the particle swarm optimization algorithm to perform elevator group control intelligent scheduling. The results show that compared with the neural network scheduling algorithm and genetic scheduling algorithm, the algorithm is dispatched and the elevator group is the value of the five performance indicators is the highest, which proves the effectiveness of the algorithm.

Introduction

In order to reduce the occupation of land and fully meet the needs of people's homes and offices, high-rise buildings have emerged in large numbers, so elevators have become one of the indispensable equipments in modern high-rise buildings. In today's world, the use of elevators has become a measure of modernity [1].The use of elevators greatly facilitates people's indoor activities, but at the same time it is also an important part of the building's internal energy consumption, which is contrary to today's energy-saving and emission reduction themes. So how to efficiently dispatch elevators to meet the passenger's basic riding index, while reducing the overall energy consumption of elevators has become a major concern in the construction industry. In this context, the elevator group control system appears, which is mainly responsible for rationally all the elevators in the building to reduce energy consumption. The core of the elevator group control system is the scheduling algorithm. With the vigorous development of artificial intelligence theory, a variety of intelligent elevator scheduling algorithms have been generated, such as fuzzy control, neural network, expert system, genetic algorithm, etc., and have received certain effects. However, these methods also have shortcomings. For example, neural network control has problems such as difficulty in determining the structure of the network itself, complicated parameter training, and difficulty in practical use. Genetic algorithms have shortcomings such as low convergence efficiency and easy precocity [2]. Therefore, the search for an efficient intelligent elevator scheduling algorithm is still an urgent problem to be solved. In this study, the finite element dynamic analysis of the elevator group control is first carried out, and then the particle swarm algorithm is used to realize the intelligent scheduling of the elevator group control system. The results show that compared with the neural network scheduling algorithm and the genetic scheduling algorithm, the elevator group control performance is greatly improved after the algorithm is scheduled.

DOI: 10.38007/Proceedings.0000850 ISBN: 978-1-80052-006-6

1 Finite element dynamic analysis of elevator group control

The basic idea of the finite element method is to discretize the continuous solution region into a finite number of combinations of cells that are joined together in a certain way. Since the cells can be combined in different ways, and the cells themselves can have different shapes, it is possible to model a complex geometry domain. Another important feature of the finite element method as a numerical analysis method is to use the approximate function assumed in each unit to slice the unknown field function to be solved on the full solution domain. The approximation function within a cell is usually expressed by the value of the unknown field function and its derivatives at each node of the cell and its interpolation function [3]. In this way, in the finite element analysis of a problem, the value of the unknown field and its derivative at each node becomes the new unknown, ie, the degree of freedom, so that a continuous finite element degree of freedom problem becomes discrete. Finite element degree of freedom problem. Once these unknowns are solved, the approximation of the field functions in each cell can be calculated by the interpolation function, and the approximate solution in the whole solution domain is obtained. Obviously, as the number of cells increases, that is, the cell size shrinks, or as the degree of freedom of the cell increases and the accuracy of the interpolation function increases, the degree of approximation of the solution will continue to improve. If the unit satisfies the convergence requirement, the approximate solution finally converges to the exact solution.

Discretizing the continuous medium into a group of units, transforming the infinite degree of freedom problem into a finite degree of freedom problem, and then solving it with a computer, the basic idea of transforming the basic idea of the finite element into a concrete realization process is the basic process of the finite element method [4].

In a nutshell, the basic idea of using the finite element method to deal with the elastic mechanics problem is to divide the solution region of the elastic body into finite elements. By constructing the interpolation displacement function and using the principle of minimum potential energy, the total potential can be used to find the linear equation. Group, thereby solving the displacement value of the unit node, and further obtaining the stress value. Specifically, it can be summarized into the following three steps:

(1) Discretization, commonly known as meshing. It discretizes a continuous elastic body that is subjected to external forces into a certain number of finite small unit assemblies. Units are only connected to each other on the nodes, that is, only nodes can transmit power.

(2) Element analysis, based on the basic equations of elastic mechanics and the variational principle, establish the relationship between the unit node force and the node displacement.

(3) Overall analysis, analysis of the overall composition of each unit. That is, the finite element equation is established according to the equilibrium condition of the nodal force, the boundary condition is introduced, the linear equations are solved, and the unit stress is calculated [5].

2 Elevator Group Control Intelligent Scheduling Algorithm Based on Particle Swarm

In 1995, American electrical engineer Kennedy and psychologist Eberhart proposed a cluster of intelligent evolutionary algorithm PSO (Particle Swarm Optimization) and used to optimize nonlinear continuous functions. The algorithm is an evolutionary algorithm that simulates the behavior of birds and fish populations. It is assumed that a group of birds are searching for food randomly and there is only one piece of food in the search area. All birds do not know where the food is, but they Knowing how far the current location is from food, the easiest and most effective way to find food is to search for nearby areas of the nearest bird [6].The particle swarm algorithm solves the optimization problem by mimicking the foraging behavior of the flock and using the cooperation of the flock.

The particle swarm algorithm is initialized into a group of random particles (random solutions), each particle has a certain fitness value (determined by the objective function), a certain speed (determining the direction and distance of the search), and the particles are optimized by iterative search. The optimal solution, and in each iteration, the particle swarm continuously follows the two optimal solutions-the individual extremum found by the particle wood and the global extremum that the entire population has searched so far. The individual extremum can be seen as the particle's own flight experience, the global extremum is the flight experience of the particle's companion, and the particle constantly updates itself according to these flight experiences [7].

2.1 Elevator group control problem description

Generally, in the building where the group control elevator is installed, each floor has an up and down call button. When the passenger arrives at a certain floor and presses the corresponding call button, the call signal is generated, and the passenger enters the car and then presses each The destination floor number, that is, the corresponding car command (stopping command).The call signal is represented by A(a1, a2, a3), where a1 is the floor number where the call signal is located, a2 is the direction of the ladder required by the call signal, and a3 is the generation time of the call signal; with B(b1, B2, b3) denotes a description of a passenger, where b1 is the initial floor at which the passenger arrives, b2 is the arrival time, and b3 is the destination floor; Use CD (cj, c2, d1, d2, ..., di) to represent the car command set of an elevator, where cj is the current floor (the corresponding floor is less when the car last stops), c2 is the car Current running direction (the running direction of the car after the last stop), di is the i-th floor stop command (i=1, 2,...,n).Have the following representation：

$$di = \begin{cases} 0, \text{The button on the i-th layer is not pressed} \\ 1, \text{The button on the i-th layer is pressed} \end{cases} \tag{1}$$

When the elevator goes up from the cj floor, $d1 = d2 = ... = dj = 0$; When the elevator descends from the cj floor, $di = dj + 1 = ... = dn = 0$.

Based on the above description, the elevator group control problem scheduling rules are as follows:

1) First-direction priority: During the dispatch process, try to assign a car that has the same direction as the outbound call request and has not passed the call floor;

2) Internal call request priority: When the destination layer request conflicts with the outbound call request, the destination layer request takes precedence, that is, the passenger is first responded to the elevator request.

3) When there are passengers in the car, the elevator must not change the running direction; when the elevator completes the last response in the running direction, if there is another call signal in the opposite direction, the elevator will run in reverse to respond to the new call. Otherwise the elevator will be idle.

4) In view of the passenger's physiological comfort and the energy consumption requirements of the elevator group control system, when the elevator is in an idle state, it should try to avoid the call response distance is less than the eleven-layer call request, when the call distance is more than two layers When inserting the outbound call signal, the number of stops of the elevator and th e congestion in the car should be reduced as much as possible.

5) The fully loaded elevator is no longer assigned, ie if the assigned elevator reaches the maximum rated load when it arrives at the call landing, no stop is given [8].

2.2 Scheduling algorithm description

The PSO algorithm first initializes a group of random particles (initial solutions) and then evolves (iteratively) to find the optimal solution. Each particle updates itself by tracking two "extreme values". One extreme value is the optimal solution found by the particle wood. This position is called the individual extremum; The other extreme is the optimal position found by the entire particle swarm, often referred to as the global extremum. The mathematical description of the PSO algorithm is:

Set in an n-dimensional search space, the group $X = \{x_1, x_2, ..., x_m\}^T$ consists of m particles, where the i-th particle is at position $x_i=\{x_{i1},x_{i2},...,x_{im}\}^T$ and the velocity is $v_i=\{v_{i1},v_{i2},...,v_{im}\}^T$.Its individual extremum is $z_i=\{z_{i1},z_{i2},...,z_{im}\}^T$ and the global extremum is $z_g=\{z_{g1},z_{g2},...,z_{gm}\}^T$. According to the principle of following the current optimal particle, the i-th particle will update its speed and position according to formula (2) and formula (3):

$$v_{id}^{t+1}=v_{id}^{t}+s_1R_1\left(z_{id}^{t}-x_{id}^{t}\right)\oplus s_2R_2\left(z_{\mathrm{gt}}^{t}-x_{id}^{t}\right) \quad (2)$$

$$x_{id}^{t+1}=x_{id}^{t}+\mathrm{v}_{id}^{t+1} \quad (3)$$

Where n is the search space dimension (the number of variables to be optimized), $d=1,2,...,n$, $i=1,2,...,m$,m is the population size; t is the current evolution algebra; s_1, s_2 is the positive acceleration constant; R_1, R_2 is the uniform distribution between 0 and 1. random number;

z_{id}^{t} represents the local optimal solution of the i-th particle found in the t-th generation; z_{gt}^{t} represents the global optimal solution of the whole population found in the t-th generation.
The algorithm flow of the standard particle swarm is shown in Fig. 1.

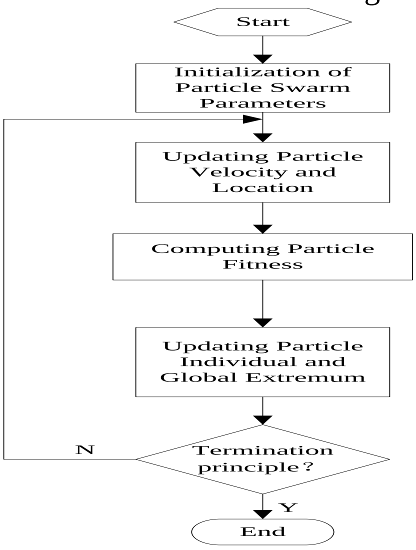

Fig 1. standard particle swarm algorithm flow

Step 1: Initialize, set parameters such as particle size, inertia weight, acceleration constant, and maximum evolution algebra;

Step 2: randomly generate the velocity and position of each particle, and set the current state of the particle to the individual extreme value, and the optimal particle in the population is set as the global extreme value;

Step 4: Compare the new fitness value of the particle with its own individual extreme value, and set the optimal fitness value as the new individual extreme value; compare the global extreme value with the new fitness value of the particle, which will be better. The fitness value is set to the new global extreme value;

Step 5: Determine whether the termination condition of the algorithm is reached. Otherwise, go to step 3. If yes, end the optimization and output the optimal solution [9].

When using the PSO algorithm to optimize the elevator group control system, each particle represents a dispatching scheme, and the dimension of the particle depends on the number of call signals currently being processed. The algorithm first randomly assigns an elevator to each call signal, calculates the fitness value of the particle according to the target optimization function, and then selects the individual extremum and the global optimal solution according to the fitness value. The system first collects the call signals of each layer (including floor information and uplink and downlink information) and the state information of the elevators, and then randomly generates the initial population, that is, the dispatching scheme, and calculates the fitness of each dispatching scheme according to the evaluation criteria of fitness. The value is selected, and the local optimal solution and the global optimal solution are selected. If the condition is satisfied, the operation ends, and the elevator is scheduled according to the dispatching scheme, otherwise the particle is updated and the next iteration is performed.

2.3 Scheduling algorithm optimization

In the elevator group control optimization scheduling, the particle swarm intelligent optimization algorithm treats the solution in the search space as the particles in the particle swarm, calculates the fitness value of each particle according to the comprehensive evaluation function, and uses the update formula to the particle swarm by iterative method. The update is performed until the optimization termination condition is reached, the global search is performed, the optimal solution is obtained, the elevator is dispatched, the elevator state is updated, and so on, until the simulation ends [10].The specific steps of the optimal scheduling strategy based on particle swarm optimization with linear decreasing inertia weights are shown in Fig. 2 below:

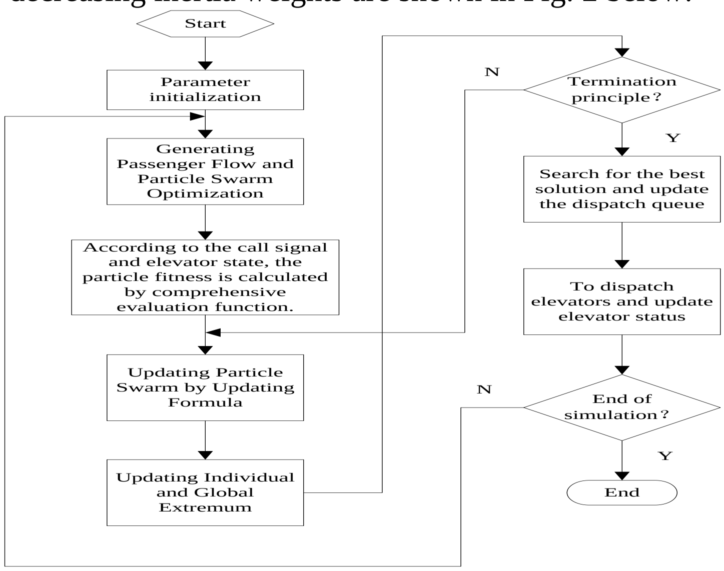

Fig 2. Is based on a particle swarm algorithm with linear decreasing inertia weights

Step 1: Initialize the elevator group control system parameters and the elevator state to generate passenger flow;

Step 2: Encoding: The encoding object is an elevator, and the encoding method used is real encoding, because in the call signal, the call signal in the floor (except the first floor and the top layer) has two states: Upstream or down, so the algorithm uses the following coding mode: In the simulation test of this paper, the elevator in the building is set to 4, so the code number 1 and 2 respectively indicate the uplink and downlink of the elevator No.1 in response to a certain floor. Calls, codes 3 and 4 respectively indicate that the second elevator responds to the uplink and downlink calls of a certain floor, and so on, so the total elevator code number is 1, 2, 3, 4, 5, 6, 7, 8 , a total of eight, exactly twice the number of elevators;

Step 3: Generate a particle group: Generate a particle group of a certain scale according to the call information. The dimension of the particle is the number of floors in the building (the floor of the simulation experiment is set to 16), and the real number in each dimension represents the number of elevators responding to the floor call. For example, the 4 in the second layer is represented by the second elevator. The downlink call signal should be layered, and 0 in the floor means no call on the floor. So particle i can be represented as $x_i = \left(z_{i1}, x_{i2}, ..., x_{iD} \right)$, and each element in the particle is an integer in the range [1,8];

Step 4: The fitness value calculation, according to the passenger's call signal and the current state of the elevator, using the comprehensive evaluation function to calculate the particle's fitness value Hik; Step 5: Use the update formula to update the particles in the particle swarm to generate a new population, and update the individual and global extremum. For the floor without the call, the fitness value will continue to maintain the set maximum value without updating. ;

Step 6: judging whether the convergence condition is reached, if not, updating the weight value, and going to step 5; if yes, performing a global search, determining the target layer, performing a dispatch, and updating the running state of the elevator;

Step 7: Determine if the simulation time is up, if not, go to step 3; if yes, the simulation ends.

3 Elevator group control system performance test

Fig. 3 below shows the parameter conditions for an elevator group control performance simulation test setup.

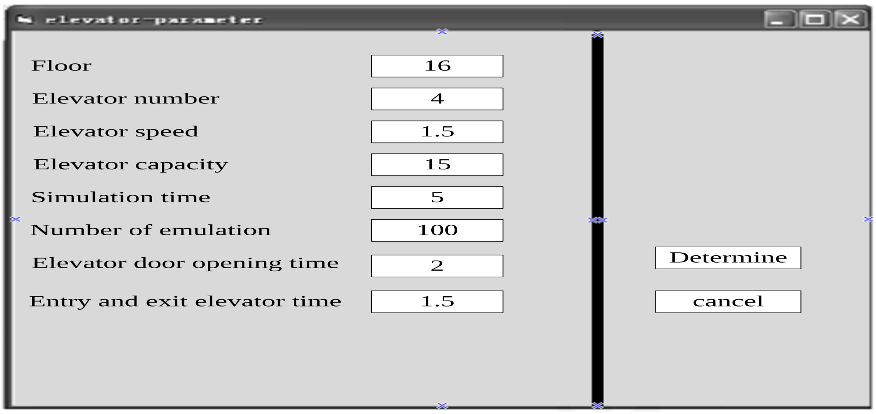

Fig 3. Parameter conditions for an elevator group control performance simulation test setup

Using the scheduling algorithm and neural network scheduling algorithm and genetic scheduling algorithm of this study, the elevator group control in a selected high-rise building is scheduled. After the scheduling is finished, the elevator group control performance test results are counted, as shown in Table 1 below.

Table 1 elevator group control performance test results

project	This scheduling algorithm	Neural Network Scheduling Algorithms	Genetic Scheduling Algorithms
Average waiting time	50.68s	68.75s	54.5s
Long Waiting Percent	3.52%	5.57%	4.68%
Maximum waiting time	88.25s	96.87s	95.68s
Number of start and stop of elevator	35 time	40 time	37time
Average Riding Time	25.89s	32.78s	28.95s

It can be seen from Table 1 that under the scheduling of the algorithm, the five performance index values of the elevator group control are higher than the other two algorithms. It can be seen that the scheduling effect of the algorithm is better.

3. Conclusion

In summary, in view of the poor performance of elevator group control system under the traditional elevator group control system scheduling algorithm, this paper studies an elevator group control artificial intelligence scheduling algorithm based on finite element dynamic analysis. In this algorithm, the finite element dynamic analysis and the optimized particle swarm artificial intelligence algorithm are used. It is verified that the proposed algorithm performs better than the neural network and genetic algorithm, and the system performance is improved, which achieves the purpose of this research. However, in the simulation test, the selected data has certain limitations, so further revision is needed.

References

[1]. Wang Qian, Zhang Cheng, Su Yang. Scheduling algorithm of elevator group control in high-rise buildings [J]. Information Recording Materials, 2017, 18 (12): 52-53.

[2]. Zhong Huilin. Elevator dispatching re-planning based on neural network [J]. Digital Technology and Application, 2016 (7): 63-64.

[3]. Jia Xuecheng, Zhu Ming, Zhang Peng, et al. Finite element analysis of impact on elevator door based on ANSYS [J]. Mechanical design and research, 2016 (3): 134-137.

[4]. Zhang Yanchao, Liu Kai, Hu Haitao, et al. Quasi-Dynamic Performances Analysis of Finger Seal Based on Finite Element Simulation [J]. Journal of Propulsion Technology, 2016, 37 (12): 2352-2358.

[5]. Hu Junfeng, Zheng Changhu, Xu Guiyang. Dynamic performance analysis of bridge compliant mechanism based on finite element method [J]. Mechanical design, 2016 (1): 81-86. Langman, Li Guoyong, Xu Chenchen. Elevator Group Control System for Energy Scheduling Optimization Simulation [J]. Computer simulation, 2017, 34 (2): 375-379.

[6]. Liu Qing, Guan Yujun. Elevator Group Control System Based on Multi-objective Programming Algorithm [J]. Journal of Hebei United university natural sclence edition, 2017 (4): 88-93.

[7]. Xiao Qing. Research on Group Control of Elevators Based on the Shortest Waiting Time [J]. Journal of Wuhan Polyiechnic, 2017, 16 (5): 106-109.

[8]. Gaodong, Xuxin. Research and implement of evaluation indexes of elevator group control algorithms and verifi-cation platform [J]. Computer Engineering and Applications, 2018 (5): 231-235.

[9]. Shi Mei. Algorithm and optimization of elevator group [J]. wuxian hulian keji, 2016 (13): 102-103.

[10]. Chagafeng. Self-optimizing strategy of multi-model elevator group control [J]. China Plant Engineering, 2017 (10): 139-140.