# Hybrid differential evolution particle swarm optimization algorithm for solving resource leveling problem of multi-project with fixed duration

Haixin Wang[1*], Shengsong Wei[1], Xin Chen[1], Mei Zhu[2], Zuhe Wang[1]
[1]College of Civil Engineering and Architecture, Shandong University of Science and Technology,
Qingdao, 266510
China
[2]Office of Registrar, Shandong TV University,
Jinan, 250014
China

Received: August 21, 2021. Revised: February 8, 2022. Accepted: February 20, 2022.
Published: March 11, 2022.

**Abstract—This paper attempts to substitute Resource Leveling Problem (RLP) into multi-project environment and construct Resource Leveling Problem of Multi-project (RLPMP) model with the goal of minimizing the sum of weighted mean square deviations of multi-resource requirements. A two-stage hybrid differential evolution particle swarm optimization algorithm is used to solve the model. In the first stage, differential evolution algorithm is used to produce new individuals, and in the second stage, particle swarm optimization algorithm uses a new speed update formula. In the first stage, in order to ensure that the optimal individual will not be destroyed by crossover and mutation, and to maintain the convergence of differential evolution algorithm, we try to introduce Elitist reservation (ER) strategy into differential evolution algorithm. In the second stage, we use a kind of Particle Swarm Optimization (PSO) algorithm with dynamic inertia weight. Through the dynamic change of inertia weight, the global search and local search ability of the algorithm can be adjusted flexibly. The case verification shows that the hybrid differential evolution particle swarm optimization algorithm can effectively solve the RLPMP model, and then effectively improve the balance of multi-project resources.**

**Keywords—Elitist reservation strategy, Hybrid differential evolution particle swarm optimization algorithm, Multi-project, Multi-resource leveling**

## I. INTRODUCTION

RESOURCE Leveling Problem (RLP) is one of the two main research issues in the field of project resource allocation, which aims to determine a project baseline schedule and minimize the fluctuation of resource utilization under the conditions of project task logic constraints and duration constraints. RLP is proved to be a strong NP-hard problem in theory and has extensive application value in practice.

Judging from the actual situation at the present stage, project management is implemented in more and more enterprises, and many enterprises in different industries usually have many different projects, and these projects often have a parallel relationship in time sequence. There is also a common demand for some bottleneck resources of enterprises. From the perspective of past practice and research, most of them carry out independent balanced optimization of project resources, which can only optimize the distribution of resources within the project, but cannot achieve the balanced allocation of resource demand in the multi-project environment of the whole enterprise [1]-[8].

Therefore, it is of more practical significance to study the problem of multi-project resource leveling. At present, some scholars have studied the resource leveling problem under the background of single project, but there are few studies on the resource balance problem of multi-project, so we try to expand the research object from a single project to multiple projects. Then build a new multi-project resource balance problem model; the solution of this model mainly includes accurate algorithm, heuristic algorithm and intelligent optimization algorithm. Among them, exact algorithms are usually suitable for exactly solving small-scale scheduling problems, but when solving large-scale problems; these algorithms often have some defects such as complex calculation and low efficiency. And the heuristic algorithms only aim at specific problems, so their versatility in practical applications is poor. In recent years, more and more researches on RLPMP turn to intelligent optimization algorithms. Because Matlab language has powerful functions such as numerical calculation, symbolic calculation, graphics and visualization, and programming, it is widely used in the research field of scientific and engineering methods of calculation, such as large-scale matrix calculation, optimization calculation, image processing, numerical calculation methods and graphical user interface design. Based on the above analysis, this paper tries to use Matlab language to choose an algorithm suitable for solving this kind of problem model.

In fact, there are many algorithms for solving this kind of problems. Among them, Particle swarm optimization (PSO) algorithm, as an evolutionary computing technology among many intelligent optimization algorithms, is more suitable for solving nonlinear, multi-peak, non-differentiable complex optimization problems, and is applicable to the solution of RLPMP However, due to the limitations of the algorithm itself, the PSO algorithm is easy to fall into local optimization when solving the optimal solution, which inevitably leads to differences between the optimized results and the expected results. Therefore, aiming at the problem of multi-project resource equilibrium optimization, this paper constructs a Resource Leveling Problem of Multi-project (RLPMP) model based on the variance of total resource consumption of all projects per unit time, and uses two-stage hybrid differential evolution particle swarm optimization algorithm to solve the model, in order to flexibly adjust the global search and local search ability of the algorithm, obtain the adjustment of the initial network graph, and achieve the goal of minimizing the variance of total resource consumption.

## II. RLPMP MATHEMATICAL MODEL

### *A. Problem description*

Assuming that there are multiple projects sharing R resources, the time parameters of each project can be obtained by calculation under the condition of the initial network diagram of each project. The goal of RLPMP is to minimize the variance of total resource consumption of all projects per unit time by adjusting the time window of non-critical processes in the network diagram, so that the demand for resources tends to be balanced gradually with the change of time. In order to make the solution of the problem operable, this paper treats a variety of resources into one resource, and the specific implementation process is as follows:

Suppose all projects have N tasks and need a total of R kinds of resources, first of all, homogenize the resources to make:

$$r_{\max}(k) = \max_i [r_0(i,k)] \tag{1}$$

After homogenizing resources, the intensity of resource demand becomes:

$$r_1(i,k) = \frac{r_0(i,k)}{r_{\max}(k)} \tag{2}$$

In (1) and (2), $i \in A$, i is the task ($i = 1, 2, ..., N$), A is the collection of project tasks, and k is the type of resources ($k = 1, 2, ..., R$). $r_0(i,k)$ indicates the demand intensity of task i for the k-th resource, that is, the demand for the k-th resource per unit time, $r_{\max}(k)$ represents the maximum demand intensity of all tasks for the k-th resource, and $r_1(i,k)$ represents the demand intensity of task i for the k-th resource after homogenization.

In this way, after homogenizing the resources, the resource intensity of each task of the project becomes a dimensionless number with a value in the range of 0-1, which makes the task comparable to each kind of resource demand, and then carries on the weighted summation of the homogenized resources. The multi-resource leveling optimization can be transformed into the single resource leveling optimization.

The following assumptions are made for the resource leveling problem: (1) Project is composed of a limited number of process tasks, and there is a fixed temporal logic relationship between tasks; (2) When the resource leveling problem is optimized, the duration of each task cannot be changed; (3) The resource requirements for each task of the project have been determined in advance and remain unchanged during the duration of the work; (4) All tasks are uninterruptible, that is, there is no interruption allowed at the beginning of the task, and there is no situation in which the task is split or segmented; (5) The actual start time of each task must be adjusted within the allowable range of total jet lag.

Mathematical model:

$$MinF=\sum_{k=1}^{K}w_k*\sigma_k^2=\frac{1}{T}\sum_{k=1}^{K}\sum_{t=1}^{T}w_k(r_k(t)-\overline{r_k})^2,\ k=1,2,...,K \quad (3)$$

Subject to:

$$r_k(t)=\sum_{i=1}^{I}\sum_{j_i=1}^{j_i}r_{k,j_i}(t)=\sum_{j_1=1}^{J_1}r_{k,j_1}(t)+\sum_{j_2=1}^{J_2}r_{k,j_2}(t)+\cdots+\sum_{j_i=1}^{J_i}r_{k,j_i}(t) \quad (4)$$

$$r_{k,j_i}(t)=\begin{cases}r_{k,j_i},TS_{j_i}\le t\le TF_{j_i}\\ 0,\quad others\end{cases} \quad (5)$$

$$S_{j_i}=TF_{j_i}-TS_{j_i} \quad (6)$$

$$ES_{j_i}\le TS_{j_i}\le LS_{j_i} \quad (7)$$

$$\max\left\{TS_{h_i}+S_{h_i}\right\}\le TS_{j_i}\le EF_{j_i} \quad (8)$$

Equation (3) is the objective function, which is used to represent the minimum sum of the weighted mean square deviation of k kinds of resource requirements. Where $w_k$ represents the weight of the k resource, T represents the project duration, $r_k(t)$ represents the quantity of the k resource needed in the t period, and $\overline{r_k}$ represents the average demand of the resource k during the construction period T;

Equation (4) represents the resource consumption of all projects on the first working day;

Equation (5) represents the resource demand intensity of the j task in the i project at time t;

Equation (6) represents the time window of task j in the i project;

Equation (7) is the time difference constraint of the actual start time;

Equation (8) represents the immediate pre-relationship constraints between project tasks.

## III. MODEL SOLVING

### A. Differential evolution algorithm

Differential Evolution (DE) algorithm is a novel heuristic intelligent search algorithm, which is originally used to solve Chebyshev multi-project problems. Stom found that it has excellent performance in solving complex optimization problems [9]. Through detailed numerical simulation experiments, it is proved that DE is an optimization algorithm with simple structure, few adjustable parameters and robustness. DE achieves population evolution through repeated iterations of selection, crossover and mutation, and then tends to global optimization. At the same time, DE is different from Genetic Algorithm (GA) in that it makes individual variation by means of the difference vector between individuals, and makes use of the characteristics of population distribution to improve the search ability of the algorithm. In addition, the greedy selection mode of one-to-one elimination mechanism is adopted in the DE selection process, which can avoid the degradation of some individuals.

The basic differential evolution algorithm is mainly composed of four processes: initialization, mutation, crossover and selection. The specific operation process is as follows:

1) Population initialization process. The initial population $P^0$ should cover the whole search area as much as possible, so if $X_i^t=(x_{i1},x_{i2},...,x_{in}),i=1,2,...NP$ is the i individual of the t iteration, in the feasible region, the initial population with population size of NP and individual component n can be generated by random sampling according to (9).

$$x_{ij}^0=rand_{ij}(0,1)(x_{ij}^{\max}-x_{ij}^{\min})+x_{ij}^{\min} \quad (9)$$

2) The process of mutation. Different from GA, the basic principle of DE mutation operation is that some difference vectors are scaled and added to another individual's base vector to get the mutated individual. According to the number of difference vectors, the calculation formula and the selection of base vectors, there are a variety of difference strategies, which are usually recorded as DE/x/y/z, in which x represents the number of basis vectors, y represents the number of difference vectors, and z represents the type of crossover operation. The main ways of variation are as follows [10]-[11].

$$DE/\mathrm{rand}/1/bin: v_i^t=x_{r1}^t+F\cdot(x_{r2}^t-x_{r3}^t) \quad (10)$$

$$DE/\text{best}/1/bin: v_i^t = x_{best} + F \cdot (x_{r1} - x_{r2}) \tag{11}$$

$$DE/\text{best}/2/bin: v_i^t = x_{best} + F \cdot [(x_{r1} - x_{r2}) + (x_{r3} - x_{r4})] \tag{12}$$

3) Crossover process. The $c_i^t$ of intermediate individuals is produced by discrete cross-recombination between the mutated individuals and the currently evolving individuals in the population, and the population diversity is increased through the interaction between the mutated individuals and the currently evolved individual elements in the population. Crossover operation refers to the use of some components of individuals in the current population and the corresponding components of mutant individuals to exchange according to certain rules to generate cross-populations. At present, the commonly used methods can be divided into binomial crossover and exponential crossover. The binomial crossover operation is relatively simple and can be realized according to (13). The binomial crossover operation produces a random decimal from 0 to 1 for each component, and if the random number is less than the crossover operator CR, it is exchanged. The larger the crossover operator coefficient is, the greater the information obtained by the crossover individual from the mutant individual is, the stronger the global search ability is and the slower the convergence is; on the contrary, the smaller the crossover operator is, the stronger the local search ability is and easy to be precocious.

$$c_{ij}^t = \begin{cases} v_{ij}^t & if\,(rand_{ij}(0,1) \le CR)\; or\; j = rand(n) \\ x_{ij}^t & otherwise \end{cases} \tag{13}$$

4) The selection process. The selection process determines the evolution direction of the population as a whole. The basic differential evolution algorithm adopts one-to-one greedy selection between the intermediate individual $c_i^t$ and the current individual $x_i^t$, that is, if the fitness of the intermediate individual is better than the current individual, the current individual is replaced, otherwise the current individual is selected. The selection process can be expressed by (14).

$$x_i^{t+1} = \begin{cases} c_i^t & if \quad f(c_i^t) \le f(x_i^t) \\ x_i^t & otherwise \end{cases} \tag{14}$$

### *B. Elitist reservation strategy*

The genes in GA do not necessarily reflect the essence of the problem to be solved, so the genes may not be independent of each other. If the genes are simply crossed, the better combinations are likely to be destroyed, so that the goal of accumulating better genes is not achieved, but the original good genes are destroyed. The elite retention strategy can prevent the optimal individual from being destroyed by the hybrid operation. Elitist reservation strategy, as a mechanism strategy of improved evolutionary algorithm, was proposed by Holland [12] in the early days. It is mainly used to deal with the problem of optimal individual loss caused by the selection error, crossover and mutation operators in the selection operator which destroy the high-order long-distance patterns. The elite individual is the individual with the highest fitness value searched by genetic algorithm so far, and it has the best gene structure and excellent characteristics. The advantage of elite retention is that in the process of evolution of genetic algorithm, the optimal individuals so far will not be lost or destroyed by selection, crossover and mutation operations. The use of elite retention strategy in the optimization algorithm can not only ensure that the individuals with better fitness values in the population will not be destroyed, but also ensure the global convergence of the algorithm [13].

### *C. Variable inertia weight particle swarm optimization algorithm*

Basic particle swarm optimization (PSO) was proposed by Clerc [14] in 1995. As a kind of parallel optimization algorithm based on swarm intelligence, PSO has been widely used in many fields. PSO algorithm is similar to GA, which starts from the random solution and evaluates the solution by fitness function through iterative optimization. Compared with GA and other algorithms, the algorithm rules of PSO are simpler and easier to implement, and the performance of algorithms such as solution accuracy and convergence speed is also better. However, its search ability and local search speed are not good enough. Therefore, in practical application, it is necessary to improve the algorithm according to the different problems and explore the solution of premature convergence to improve the efficiency of the algorithm.

In the standard particle swarm optimization algorithm, the search space is J-dimensional and the number of particles is N. Then the position of the i-th particle can be expressed as the velocity corresponding to the $x_i = (x_{i1}, x_{i2}, \cdots, x_{iJ}), i = 1, 2, \cdots N$; The corresponding velocity of each particle is expressed as $v_i = (v_{i1}, v_{i2}, \cdots, v_{iJ}), i = 1, 2, \cdots N$. There are usually two factors to consider when searching for each particle, one is the optimal value $p_i$, pi=1 found by individual traversal search, the other is the optimal value pg, pg=1 found by all particles traversal search, which can be marked as gb.

In the standard particle swarm optimization algorithm, the search space is J-dimensional and the number of particles is N. Then the position of the I-th particle can be expressed as the velocity corresponding to each particle in $x_i = (x_{i1}, x_{i2}, \cdots, x_{iJ}), i = 1,2,\cdots N$; the corresponding velocity of each particle is expressed as $v_i = (v_{i1}, v_{i2}, \cdots, v_{iJ}), i = 1,2,\cdots N$. There are usually two factors to be considered when searching for each particle, One is the optimal value found by traversing the individual $p_i$, $p_i = (p_{i1}, p_{i2}, \cdots, p_{iJ}), i = 1,2,\cdots N$, which can be marked as $P_{best}$; the other is the optimal value found by traversing all the particles $p_g$, $p_g = (p_{g1}, p_{g2}, \cdots p_{gJ}), g = 1,2,\cdots N$, which can be marked as $G_{best}$.

Update the speed and position of particles according to (15) and (16):

$$v_{ij}(t+1) = v_{ij}(t) + c_1 \cdot rand_1() \cdot [P_{ij}(t) - x_{ij}(t)] + c_2 \cdot rand_2() \cdot [P_{gj}(t) - x_{ij}(t)] \tag{15}$$

$$x_{ij}(t+1) = v_{ij}(t+1) + x_{ij}(t) \tag{16}$$

After the analysis of the general problem [13], the law is summarized, that is, in the process of solving the general problem, the distribution of particles at the beginning of the iterative process is relatively dispersed, and it is necessary to use a strong global search ability to make the particles quickly converge to the optimal solution, but when the algorithm is implemented in the later stage, after the particles converge to a certain area near the optimal solution, it is necessary to reduce the speed to conduct a detailed local search. Therefore, although the inertia weight coefficient is taken as a constant at the initial stage, through experiments, scholars find that dynamic $\omega$ can obtain better optimization results than fixed value $\omega$.

In order to better coordinate the balance between global search ability and local search ability of particle swarm optimization algorithm in order to control its development and optimization detection ability, a strategy of Linearly Decreasing Weight (LDW) is proposed [15]. By using changeable weight, a larger $\omega$ value is set at the beginning, and gradually decreases with iteration, thus gradually turning from exploration to discovery. Experiments have proved that LDW strategy can effectively improve the optimization effect of PSO.

A Logarithmic Decreasing Weight (LOGW) is proposed [16], that is:

$$\omega = \omega_{\max} - \alpha \cdot (\omega_{\max} - \omega_{\min}) \cdot \log_{T_{\max}}^{Iter} \tag{17}$$

It is proposed that $\alpha$ is a logarithmic adjustment factor, which is called a logarithmic compression factor when $0 < \alpha < 1$ and a logarithmic expansion factor when $\alpha > 1$. Through the test experiment, it is found that the performance of LOGWPSO algorithm is obviously better than that of LINWPSO.

In order to change the single adjustment mode of LINW strategy, the concept of change rate of focus distance is introduced [17], which is defined as:

$$k = \frac{MaxDist - MeanDist}{MaxDist} \tag{18}$$

Among them, $MaxDist$ is the maximum focusing distance and $MeanDist$ is the average focusing distance.

$$MeanDist = \frac{\sum_{i=1}^{m} \sqrt{\sum_{d=1}^{D} (p_{td} - x_{id})^2}}{m} \tag{19}$$

$$MaxDist = \max_{i=1,2,..,m} \sqrt{\sum_{d=1}^{D} (p_{td} - x_{id})^2} \tag{20}$$

Where m is the number of particles in the particle swarm, D is the dimension of each particle, $p_{td}$ is the optimal position searched by the particle swarm, and $x_{id}$ is the optimal position searched by each particle. Based on the adaptive inertia weight of k, it is judged that the particle should improve its global search ability or local search ability by calculating the value of k in each iteration, and then adjust the inertia weight dynamically. The experimental results show that the convergence speed and accuracy of the particle swarm optimization algorithm with dynamical inertia weight (DCWPSO) are better than those of LINWPSO. Therefore, this paper uses this method to flexibly adjust the global search and local search ability of the two-stage hybrid algorithm through the dynamic change of inertia weight.

### *D. Algorithm design*

1) Encoding method

The topological structure in the Real-number Encoding (RE) formal space is consistent with the topological structure in the expressive space, which can improve the efficiency of the algorithm. In general, real number coding is carried out according to the variables in the objective function. Combined with the characteristics of RLPMP, the real number coding based on the actual start time of all tasks in each project can simplify the complexity of the problem. In order to further simplify the problem-solving process, this paper only codes the actual start time of tasks on non-critical lines.

2) Fitness function

The optimization goal of RLPMP is to minimize the variance of total resource consumption $\sigma^2$, which belongs to the minimization problem, so it is necessary to map the original objective function into the fitness function $f(i)$ in the form of maximum value to ensure that the appropriate individual has a larger fitness value.

If $fitness(i)$ is the fitness value of the $i$ chromosome of the current population, the fitness function can be expressed as follows:

$$fitness(i) = \frac{C}{F} \tag{21}$$

Where C is a non-zero non-negative real number and F is the value of the RLPMP objective function.

3) Treatment of constraint conditions

Penalty function method is a common method of constrained optimization, and its basic idea is to construct constraint function with constraint conditions and transform the constraint problem into an unconstrained problem [18]. Because the outer point function method can not only solve the optimization problems with equality constraints and inequality constraints at the same time, but also has no regional restrictions in the construction of penalty function and the selection of initial points, it is convenient for practical calculation. Therefore, this paper chooses the external point penalty function method as the constraint treatment method.

In view of the inequality constraint form $g_i(x) \le 0$ of the problem model in this paper, the penalty function can be expressed as follows:

$$F(x, M^{(k)}) = f(x) + M^{(k)} \sum_{i=1}^{m} \{\max[g_i(x), 0]\}^{\eta} \tag{22}$$

Among them, the penalties are:

$$\max[g_i(x), 0] = \frac{g_i(x) + |g_i(x)|}{2} = \begin{cases} g_i(x) & , g_i(x) > 0 \\ 0 & , g_i(x) \le 0 \end{cases} \tag{23}$$

That is, when the search point $x^{(k)}$ is in the feasible domain, the penalty term is zero; when it is no longer the feasible domain, the penalty term is non-zero and increases with the increase of $M^{(k)}$. Therefore, if the value of the augmented objective function is minimized, it is necessary to make the penalty term zero, that is, to satisfy the constraint condition $g_i(x^{(k)}) \le 0$.

### *E. The procedure of DE-DCWPSO algorithm*

*1) The two-stage of DE-DCWPSO algorithm*

As an evolutionary computing technology among many intelligent optimization algorithms, particle swarm optimization (PSO) is more suitable for solving nonlinear, multi-peak and non-differentiable complex optimization problems, and is suitable for solving the RLPMP model constructed in this paper. However, due to the limitations of the algorithm itself, the PSO algorithm is easy to fall into local optimization when solving the optimal solution, which inevitably leads to the difference between the optimization results and the expected results. Therefore, this paper attempts to use two-stage hybrid differential evolution particle swarm optimization algorithm to solve the model. In the first stage, differential evolution algorithm is used for evolution. in order to ensure that the optimal individual is not destroyed by crossover, mutation and other operations, and to maintain the convergence of differential evolution algorithm, we try to introduce elitist reservation (ER) strategy into differential evolution algorithm to get a new individual $x_{ide}$; In the second stage, the particle swarm optimization algorithm is adopted, and the new individual $x_{ide}$ generated in the first stage is introduced to generate a new speed update (24):

$$v_{ij}(t+1) = v_{ij}(t) + c_1 \cdot rand_1() \cdot [x_{ide}(t) - x_{ij}(t)] + c_2 \cdot rand_2() \cdot [P_{gj}(t) - x_{ij}(t)] \tag{24}$$

The individual $x_{ide}$ obtained in the first stage is used to replace the individual optimal Pi in the original speed update (15), and then a kind of PSO algorithm with dynamic inertia weight is used to flexibly adjust the global search and local search ability of the algorithm by dynamically changing the inertia weight. Finally, the improved two-stage hybrid differential evolution particle swarm optimization algorithm is applied to solve the RLPMP model, and a better optimization result is expected.

*2) The specific procedure of DE-DCWPSO algorithm as follows*:

a) Setting up the relevant parameters of DE-DCWPSO algorithm and iteration times t=0; Setting up max iteration times Maxit.

b) Initializing population randomly, computing the fitness value, solving the optimal solution Pi of individuality and the optimal solution Pg in the whole population.

c) The two sub-population of DE algorithm carry on variation respectively according to (11) and (12). Carrying on crossover according to (13). Finally, carrying on selection and obtaining new population according to (14).

d) Updating speed and position for PSO population according to (16) and (24).

e) Computing new fitness value after the operation of PSO evolution, updating Pi and Pg.

f) Updating iteration times t=t+1, if attaining the max iteration times, terminating algorithm, outputting optimal solution; Otherwise, turning to the c procedure.

## IV. CASE ANALYSIS

As shown in Fig. 1, this paper uses two parallel projects as the target case, the project I and project II contain is a project with 10 tasks respectively. Letter in task node represents the task serial number, number above task node expresses the task duration and number below task node indicates respectively the quantities that the task in unit time consumes three kinds of updatable resources. Afterwards, this paper computes the resource importance. Then, the paper unifies sources and setting the weight coefficient of each resources and the multi-resource leveling optimization were transformed as the single resource leveling optimization [19]. Because determining the weight coefficient of resources is not the research focus of this paper and the importance of each resource is different in different project respectively, the resource weight coefficients are respectively determined as W1=0.2, W2=0.5, andW3=0.3 without loss of generality and under the constraint condition $w_k \leq 1, \sum_{k=1}^{r} w_k = 1$.

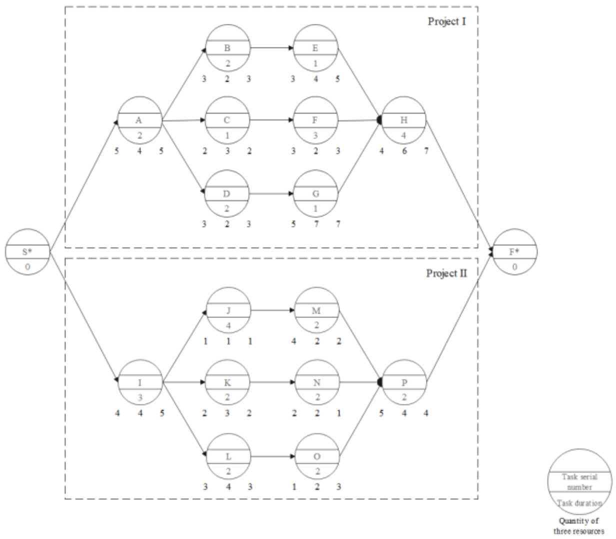



Fig. 1 The AON network after the merger of the two projects

This paper solves the mathematical model of case based on the two-stage DE-DCWPSO algorithm and the algorithmic parameters are respectively set as the population size popsize =20; the crossover probability Itermax=100; the scaling factor F=0.9. The comparison of average daily resource consumption of before and after optimization is shown in Fig. 2 and Fig. 3. The start and end time of each task after two projects resource leveling optimization is specifically shown in Table I.

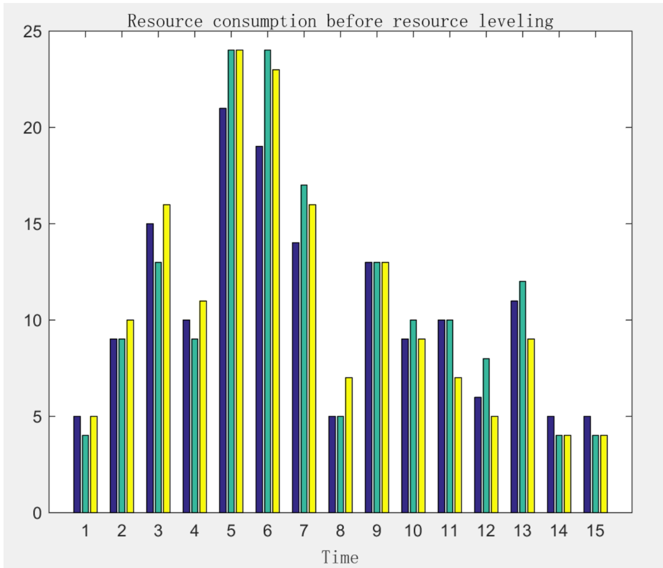
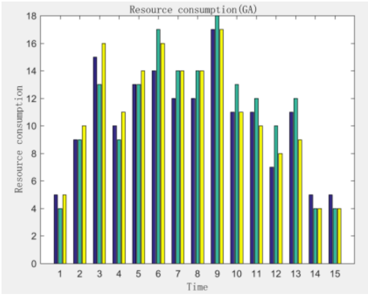
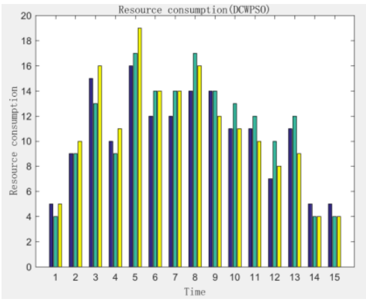


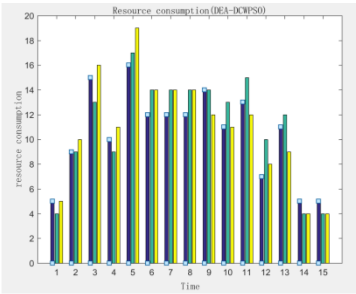

Fig. 2 Histogram of daily average resource consumption before resource leveling




(a)




(b)




(c)

Fig. 3 Histogram of daily average resource consumption after resource leveling

Table I Start and end time of each project task

| Task number | Start time | End time | Project number | Task number | Start time | End time | Project number |
|---|---|---|---|---|---|---|---|
| A | 3 | 5 | 1 | I | 4 | 7 | 2 |
| B | 5 | 7 | 1 | J | 7 | 11 | 2 |
| C | 7 | 8 | 1 | K | 11 | 13 | 2 |
| D | 5 | 7 | 1 | L | 8 | 10 | 2 |
| E | 7 | 8 | 1 | M | 11 | 13 | 2 |
| F | 8 | 11 | 1 | N | 13 | 15 | 2 |
| G | 9 | 10 | 1 | O | 12 | 14 | 2 |
| H | 11 | 15 | 1 | P | 15 | 17 | 2 |

From the operation results of algorithm, the variance is 38.5238 before equalization and converges to 17.3238 after iteration of the DE-DCWPSO algorithm, which has achieved good equalization effect. Through comparing the histogram of average daily resource consumption of before and after equalization, we can see that the optimized multi-project resource consumption is more balanced. To further verify the algorithm that used in this paper, this paper uses GA and DCWPSO algorithm to solve the RLPMP mathematical model and GA algorithmic parameters are set as the maximum number of iterations itermax=50; the population number popsize=20; the mutation rate PM=0.1; the crossover rate PC=0.6; Finally, GA algorithm converges to the optimal value 18.1524. DCWPSO algorithmic parameters are set as speed update parameter $c_1 = 1.8$, $c_2 =,1.4$; maximum inertia factor InertiaMax=0.93; minimum inertia factor InertiaMin=0.5; the number of iterations ITmaxgen=50; particles swarm scale popsize=20. Finally, DCWPSO algorithm converges to the optimal value 18.0592. Therefore, compared to GA and DCWPSO, DE-DCWPSO can ensure that individuals with better fitness values are preserved without being destroyed in population and ensure the global convergence of algorithm by a case. In the iteration of algorithm, DE-DCWPSO algorithm with two-stage optimization reduces the premature convergence of algorithm and improves the algorithmic efficiency and the equalization effect is better. The comparison of three algorithmic equalization effect is shown in Table II.

Table II The actual start time of each task and its resource variance

| Plan | Start plan | GA | DCWPSO | DE-DCWPSO |
|---|---|---|---|---|
| A | 3 | 3 | 3 | 3 |
| B | 5 | 5 | 5 | 5 |
| C | 7 | 7 | 7 | 7 |
| D | 5 | 6 | 5 | 5 |
| E | 7 | 7 | 7 | 7 |
| F | 8 | 8 | 8 | 8 |
| G | 8 | 9 | 9 | 9 |
| H | 11 | 11 | 11 | 11 |
| I | 4 | 4 | 4 | 4 |
| J | 7 | 7 | 7 | 7 |
| K | 7 | 11 | 10 | 11 |
| L | 7 | 8 | 8 | 8 |
| M | 11 | 11 | 11 | 11 |
| N | 10 | 13 | 13 | 13 |
| O | 13 | 12 | 12 | 12 |
| P | 15 | 15 | 15 | 15 |

| | | | | |
|---|---|---|---|---|
| Variance | 38.5238 | 18.1524 | 18.0952 | 17.3238 |

## V. Conclusion

RLP is one of the two key issues in the optimal allocation of resources during project implementation. According to the multi-project and multi-resource environment faced by related enterprises in the process of project management, this paper develops original RLPMP model with the minimum of sum of the weighted mean square deviation of the multi-resource demand as the optimization goal. This paper tries to use a two-stage hybrid differential evolution particle swarm algorithm to solve the model that has been constructed. This paper uses differential evolution algorithm to produce new individuals in the first stage and the particle swarm algorithm uses new speed update formula in the second stage. In the first stage, in order to ensure that the optimal individuals will not be destroyed by operations such as crossover and mutation and maintain the convergence of differential evolution algorithm, this paper tries to introduce the strategy of elite reservation into the differential evolution algorithm. In the second stage, this paper uses a kind of PSO algorithm with dynamic inertia weight and flexibly adjusts the algorithmic capabilities of global and local search. The effectiveness of the DE-DCWPSO algorithm for solving RLPMP is further verified by a case and the algorithmic results are compared with the results of GA and DCWPSO algorithm. The results show that the DE-DCWPSO algorithm can more effectively improve multi-project resource balance and then provide a preference for the balanced allocation of resources of relevant enterprises during multi-project implementation. Our next plan is to further improve and build a new multi-objective optimization model according to the real background of RLPMP in the process of multi-project implementation; in addition, our existing algorithm is only suitable for solving single-objective optimization problems, so we will try to use multi-objective optimization algorithms to solve the new model.

## Acknowledgment

The research was sponsored by the Initial Scientific Research Fund in the Shandong University of Science and Technology [grant number 2019RCJJ026], Shandong Provincial Key Research and Development Program of China (Soft Science program) [grant number 2021RKY06105] and Education and Teaching Research Program of Shandong Province (General Program) [grant number 2021JXY076]. The authors are grateful for their support. The contribution of this research was based on previous studies, and the authors of this paper express their heartfelt thanks to the authors of all listed previous works.

## References

[1] S.S. Leu, C.H. Yang, “GA-based multi-criteria optimal model for construction scheduling,” Journal of Construction Engineering and Management, vol. 125, pp. 420–427, 1999.

[2] A. Hadeel, H. Moncer, “Hybrid meta-heuristic methods for the multi-resource leveling problem with activity splitting,” Automation in Construction, vol. 27, pp. 89-98, 2012.

[3] P. Ghoddousi, E. Eshtehardian, S. Jooybanpour, and A. Javanmardi, “Multi-mode resource-constrained discrete time-cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm,” Automation in Construction, vol. 30, pp. 216-227, 2013.

[4] A. Damci, D. Arditi, and G. Polat, “Impacts of different objective functions on resource leveling in construction projects: A case study,” Journal of Civil Engineering and Management, vol. 20, pp. 37-547, 2014.

[5] M. T. Zhang, M. Yu, and F. W. Kong, “Resource balance of prefabricated building project based on genetic algorithm,” Journal of Civil Engineering and Management, vol. 37, pp. 69-175, 2020.

[6] Y.P. Li, Y.F. Li, Z.L. Sun, and H. Guan, “Optimization method of engineering project resource balance based on SAGA,” Journal of Civil Engineering and Management, vol. 38, pp. 4-94, 2021.

[7] J. Wang, K.X. Liu, X.Q. Zhang, and T. Chen, “Optimization algorithm for resource leveling of construction projects with multiple resources based on subset simulation,” Journal of Hunan University (Natural Sciences), vol. 48, pp. 8-176, 2021.

[8] C.G. Provatidis, “Teaching the Fixed Spinning Top Using Four Alternative Formulations,” WSEAS Transactions on Advances in Engineering Education, vol. 18, pp. 0-95, 2021.

[9] R. Stom, K. Price, “Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, pp. 341-359, 1997.

[10] J. Brest, B. Boskovic, S. Greiner, V. Zumer, and M. S. Maucec, "Performance comparison of self-adaptive and adaptive differential evolution algorithms," Soft Computing, vol. 11, pp. 617-629, 2007.

[11] S. Das, A. Abraham, U.K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood based mutation operator," IEEE Transactions on Evolutionary Computation, vol. 13, pp. 526-553, 2009.

[12] H. Holland, "Adaptation in natural and artificial system," Ann Arbor: The University of Michigan Press, MI, 1975.

[13] A. Colorni, M. Dorigo, V. Maniezzo. Distributed optimization by ant colonies. The First European conference on Artificial Life. France: Elsevier, 1991, pp. 134-142.

[14] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," Proceedings of Congress on Evolutionary Computation, 1999, pp. 1951-1957.

[15] Y. Shi. RC. Eberhart, "Empirical study of particle swarm optimization," Proceedings of the Congress on Evolutionary Computation. Piscataway: IEEE Service Center, 1999, pp. 1945-1950.

[16] W.Z. Dai, X.L. Yang. "Particle swarm optimization algorithm based on inertia weight logarithmic decreasing," Computer Engineering and Applications, vol. 51, pp. 14-19, 2015.

[17] Z.H. Ren, J. Wang. "New adaptive particle swarm optimization algorithm with dynamically changing inertia weigh," Computer Science, vol. 36, pp. 227-229, 2009.

[18] Y. Li, "Application of penalty function method in machine element design," Guang Zhou: South China University of Technology, 1985.

[19] Z.H. Wang, X. Qi. "The weight optimal choice method of multi-resource leveling," Journal of Industrial Engineering Management, vol. 16, pp. 91-93, 2002.

The lead author of this paper obtained a doctorate in management science and engineering in 2017. The degree is awarded by Shandong University of science and technology in Qingdao, China. At present, his main research field is the research on project resource allocation and optimization methods.

He used to work in China Construction Installation Group Co., Ltd. and Shandong University of architecture, and now works in the Department of engineering management, School of civil engineering and architecture, Shandong University of science and technology. He has always been committed to the research on enterprise project resource allocation and optimization methods. Now he mainly studies enterprise multi-project resource allocation and optimization methods.

Dr. Wang is currently a member of the Chinese Architectural Society.

**Author Contributions**

In this paper, Haixin Wang developed the research and organized the research flow; Haixin Wang, Mei Zhu, Xin Chen and Shengsong Wei implemented the research program and collected data; Mei Zhu, Zuhe Wang and Xin Chen participated in the case study analysis; and H.W. completed the writing. All authors were involved in the preparation and validation of the manuscript, and have read and agree to the published version of the manuscript.