J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 8701--8717

Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Simulation Optimisation of Laboratory Resources in Universities with Intelligent Scheduling Strategies

Chao He[1], Shi Cheng[2,✉]

[1] Asset and Laboratory Management Department, Nantong University Xinglin College, Nantong, Jiangsu, 226000, China
[2] School of Artificial Intelligence and Computer Science, Nantong University, Nantong, Jiangsu, 226000, China

ABSTRACT
This paper establishes a solution model for resource scheduling optimization in university laboratories, and sets the corresponding constraints and objective functions. The genetic algorithm under the heuristic algorithm is used to solve the resource scheduling optimization problem. On this basis, the pyramid model is constructed, the population evolution and variant strategy are proposed respectively, the model genes are labeled with scheduling cost adaptation, and the genes are generated in series. The framework of scheduling algorithm is proposed, and the dynamic scheduler is constructed to realize the scheduling of university laboratory resources. Through simulation experiments and algorithm analysis, the effectiveness of the use of the model is verified. The experimental results show that when the number of simulation is 10 times, the fitness of the population is 20, 100 and 200 respectively. After the implementation of scheduling for college laboratory resources, the utilization rate of laboratory equipment is increased by 16.3%, 34.6% and 18.4% respectively.

Keywords: scheduling strategy, constraints, genetic algorithm, pyramid model, laboratory resources

1. Introduction

With the rapid development of computer information technology, the increasing number of users in college laboratories has led to the sharing of laboratory hardware and software resources and personalized contradictions gradually prominent. In order to meet the demand for high resource utilization and batch configuration of personalized experimental environments for hardware and software equipment in college laboratories, the construction of digital experimental platforms as well as intelligent configuration of resources is very critical [1-2].

Cloud computing can meet the large-scale computing needs, which virtualizes a large number of distributed computer hardware and software resources into a huge pool of shared resources through

✉ Corresponding author.
E-mail address: chengshi@email.cn (S. Cheng).
Received 05 March 2024; Revised 20 April 2024; Accepted 05 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-482

the Internet, so that users can use the relevant cloud services provided by cloud computing at any time and any place through the network on demand [3-5]. The cloud computing digital experiment platform treats the experimental resource demand proposed by the user as a specific experimental task, and implements experimental task allocation after forming a task resource pool according to the priority level [6-8]. It is also able to re-divide the task resource pool into multiple function-specific resource pools according to different functions to achieve unified scheduling of experimental resources [9-10]. The experimental resources of cloud computing-based digital experimental platform can realize resource expansion at any time according to the configuration of laboratory equipment in colleges and universities, and the experimental resources within college disciplines present interactive sharing, and the experimental resources between disciplines and disciplines can also realize interactive sharing [11-14]. Therefore, this kind of digital experiment platform has good openness and expandability, which really improves the utilization efficiency of experimental resources.

Related scholars have conducted a lot of research on how to achieve the scheduling and management of university laboratory resources in cloud environment.Arunarani, A. R. et al. showed that the principle of scheduling virtualized resources in cloud computing management is to minimize the time loss and maximize the performance, discussed the problems and obstacles that need to be cleared under different task mobilization strategies, and looked forward to the future research [15]. Domanal, S. G. et al. compared the new hybrid bionic algorithm with the traditional scheduling algorithm in the study of resource allocation and management in cloud computing systems and found that the hybrid algorithm has higher utilization and less average response time [16]. Strumberger, I. et al. proposed swarm intelligent meta-heuristic hybrid whale algorithm to enhance the resource scheduling performance in cloud systems and simulation experiments the results show that the average performance of the proposed algorithm outperforms the original algorithm as well as other heuristic and meta-heuristic algorithms [17]. Arabnejad, V. et al. investigated the performance of Proportional Deadline Constrained Algorithm (PDC) and Deadline Constrained Critical Path Algorithm (DCCP) on cloud resource workflow scheduling problem and found that both algorithms have higher cost efficiency and success rate [18]. Gai, K. et al. combined information-physical fusion system with heterogeneous cloud computing and proposed an optimized workload model based on intelligent cloud, which utilizes the predicted cloud capacity and considers sustainable factors to optimize the task allocation objectives of heterogeneous cloud [19]. Casas, I. et al. based on balanced file reuse-replica scheduling algorithm for cloud computing environment to perform the workflow of scientific application Optimized scheduling, which not only parallelizes scientific workflows, but also optimizes the transfer process of resource tasks using data reuse and replication techniques [20].

In this paper, the constraints and objective conditions set by the model are defined respectively according to the needs of intelligent scheduling of laboratory resources in universities. Based on the genetic algorithm, the population evolution strategy of the pyramid model is proposed, and the random selection form of roulette method is used to generate gene strings for each gene within the model according to the scheduling cost labeling adaptation. For the parallel evolution of the bottom population, provide cyclic conditions for the new population, and based on the population size and other conditions, divide the population, screen out the chromosomes with better adaptation in the population, merge the high-quality chromosomes, and form a new population to further realize the optimization of the algorithm. Construct the algorithm framework for scheduling resources in college laboratories, and propose the dynamic scheduler method to realize the reasonable scheduling of resources in college laboratories. Simulation experiments and empirical analysis method are used to analyze the application effect of intelligent scheduling strategy.

2. Modeling and Optimization of Laboratory Resource Scheduling Strategies in Higher Education Institutions

2.1. Modeling

2.1.1. Description of the research problem. The main resources involved in the teaching arrangement of laboratory courses in higher education are time, courses, classes, laboratories, teachers, etc., while the resource scheduling problem in examination arrangement is the problem of scheduling a series of examinations in the course time slots and laboratories which have been defined in advance, and at the same time satisfying all the basic constraints, hard constraints and trying to satisfy as much as possible the soft constraints. The basic and hard constraints need to be satisfied absolutely, while the soft constraints need to be satisfied as much as possible. Soft constraint satisfaction is an important criterion for evaluating the quality of course scheduling solutions. The more soft constraints are satisfied, the more effective the course scheduling is. How to schedule all the courses in the right time and laboratory is the focus of course scheduling problem research.

In this paper, we are looking for an effective way to solve the laboratory course scheduling problem quickly, accurately and efficiently by using an excellent intelligent algorithm to produce an excellent solution that meets the requirements of the teaching task and maximizes the use of resources. An excellent course scheduling solution can effectively reduce the waste of teaching resources, save time, save manpower, so as to optimize the teaching work and improve the quality of teaching.

2.1.2. Description of symbols and definition of meanings. Combined with the previous summary and analysis of the laboratory foundation course examination arrangement of the main resources involved in the time, course, class, examination room, teacher, now on the relevant resources set of representative symbols to explain, involved in the definition of the function.

1) Resource set representative symbols

(1) Time Collection: $T=\left\{t_l,t_2,\dots,t_h,\dots,t_H\right\}$, for a total of H exam time slots, $H=|T|$.

(2) Course Pool: $C=\left\{c_1,c_2,\dots,c_n,\dots,c_N\right\}$, a total of N courses taking the exam, $N=|C|$.

(3) Class collection: $G=\left\{g_1,g_2,\dots,g_i,\dots,g_t\right\}$, total I classes taking the exam, $I=|G|$.

(4) Pooling of required equipment: $E=\left\{e_1,e_2,\dots,e_j\dots,e_J\right\}$, for a total of J pieces of equipment, $J=|E|$.

Priority is given to placing larger classes in labs with more equipment for exams.

$Cap\left(e_j\right)$ denotes the size of the capacity of the j nd lab, then $Cap\left(e_1\right)\geq Cap\left(e_2\right)\dots\geq Cap\left(e_J\right)$.

(5) Teacher set: $M=\left\{m_l,m_2,\dots,m_k,\dots,m_k\right\}$, a total of K teachers to participate in the teaching work, $K=|M|$.

Combined with the number of classes in the university now set the way as well as the course to the class as a unit for the arrangement of the custom, generally each laboratory is arranged for two teachers, so for the definition of the teacher group set is MM, arranged in the classroom time t_h of the teacher group set is defined as MMt_h:

$$MM=\left\{mm/mm=\left(m_im_j\right)m_i,m_j\in M,i\neq j\right\} \tag{1}$$

MMt_h is a subset of the set of teacher groups MM, i.e., $MMt_h\subset MM$. Since teacher m_i cannot teach two or more classes at the same time during class time t_h, teacher m_i can only belong to an element of MMt_h, which leads to $|MMt_h| \le |M|/2$. Assume $|MMt_h| = P$ here.

(6) The set of labs for lab course c_n:

$$CE_{c_n} = \left\{ ce_{c_n} \middle| ce_{c_n} \le \frac{x(c_n)}{c_n} ce_{c_n} \in X \right\} \tag{2}$$

(7) Students have completed their course selection before the start of the semester's experimental courses, and the school's experimental courses are selected on a class-by-class basis, so the relationship between the course and the students is determined before scheduling the experimental course, which means that the course and the class are correlated, and scheduling the course is also scheduling the class. Define this as a lab course task, a collection of tasks:

$$ET = \left\{ et \middle| et = (c_n, g_i) c_n \in C, g_i \in G \right\} \tag{3}$$

where $et = (c_n, g_i)$ denotes that the students in the g_i class are enrolled in the c_n course.

(8) A collection of lab course scheduling unit tables:

$$ES = \left\{ \begin{array}{l} es \middle| es = (c_n, g_i, e_j, mm_k, t_h) c_n \in C \\ g_i \in G, e_j \in E, mm_k \in MM_{t_h}, t_h \in T \end{array} \right\} \tag{4}$$

where $es = (c_n, g_i, e_j, mm_k, t_h)$ denotes that a student in the g_i class is in the e_j laboratory at time t_h for a c_n course taught by a group of teachers mm_k conducting the course.

2) Definition of functions

The functions that will be used in the specific study of this paper are defined below.

(1) $X(c_n)$: Indicates the number of classes participating in the c_n experimental course.

(2) $X(g_i)$: Indicates the number of subjects to be taken in the laboratory course by the students in the class g_i.

(3) $X(t_h, e_j)$: Indicates the number of students in the laboratory e_j at the time of class t_h hours.

$$T(m_k, t_h) = \begin{cases} 1 & Invigilator\ m_k\ at\ t_h\ time\ without\ special\ circums\tan ces \\ 0 & Invigilator\ m_k\ at\ t_h\ time\ with\ special\ circums\tan ces \end{cases} \tag{5}$$

2.1.3. Definition of constraint functions. The next step will be to define a function for the constraints set above, preceded by the definition of the associated function [21]:

$$x_{g_i t_h e_j c_n mm_{n_p}} = \begin{cases} 1 & \begin{array}{l} Students\ of\ class\ g_i\ appearing \\ for\ the\ exa\min ation\ of\ course \\ cn\ invigilated\ by\ the\ group\ of \\ invigilating\ teachers\ of\ mm_{n_p} \\ at\ time\ t_h\ at\ the\ e_j\ exa\min ation\ centre \end{array} \\ 0 & Otherwise \end{cases} \tag{6}$$

$$y_{c_n t_n} = \begin{cases} 1 & Courses\ c_n\ be\ scheduled\ for\ exams\ at\ time\ t_h \\ 0 & Otherwise \end{cases} \tag{7}$$

$$z_{m_k t_h}\begin{cases}1 & \begin{array}{c}\textit{Invigilator } m_k \textit{ is scheduled to} \\ \textit{invigilate the exam at time } t_h\end{array} \\ 0 & \textit{Otherwise}\end{cases} \tag{8}$$

(1) Each student in the same class cannot be scheduled for two or more lab courses at the same time during the same session:

$$M_1:\sum_{P=1}^{p}\sum_{n=1}^{N}\sum_{j=1}^{J}x_{g_i t_n e_j c_n m m_{n_p}} \le 1\left(i=1,2,\ldots,I \quad h=1,2,\ldots,H\right) \tag{9}$$

(2) Only one class may be taught by the same teacher (a uniform group of teachers) during the same period:

$$M_2:\sum_{i=1}^{L}\sum_{j=1}^{J}\sum_{n=1}^{N}x_{g_i t_h e_j c_m n_{np}} \le 1\left(p=1,2,\ldots,p=1,2,\ldots,H\right) \tag{10}$$

(3) All Yes to Take courses are scheduled for class periods:

$$M_3:\sum_{h}^{H}\sum_{n}^{N}y_{c_n t_h} = \left|C\right| = N \tag{11}$$

(4) All students from all classes taking the same laboratory course must be present at the same time when attending that class session:

$$M_4:\sum_{h}^{H}y_{c_n t_h} = 1\left(n=12,\ldots,N\right) \tag{12}$$

(5) The number of students participating in a laboratory session in each examination room must not exceed the capacity of the laboratory:

$$M_5:X\left(t_h,e_j\right)\le Cap\left(e_j\right)\left(h=1,2,\ldots,H \quad j=1,2,\ldots,J\right) \tag{13}$$

(6) Not scheduling laboratory course instructors at times when they have special circumstances:

$$M_6:z_{m_k t_h}=1 \ And\ T\left(m_k,t_h\right)=1\left(h=1,2,\ldots,H \quad k=1,2,\ldots,K\right) \tag{14}$$

2.1.4. Definition of the objective function. In experimental course examination arrangement, basic constraints and hard constraints should be absolutely satisfied, while soft constraints need to be satisfied as much as possible. Soft constraint satisfaction is an important criterion for evaluating the quality of laboratory course arrangement solutions. The more soft constraints are satisfied, the better the experimental course arrangement is. Therefore, soft constraints are used as the objective function [22].

1) Try to ensure a reasonable time between two adjacent lab sessions in the same class, so that students have enough time to preview and combine the difficult and easy. At the same time the class whose other courses end early conducts the lab session first.

Laboratory course time optimization scheduling function: f_1

2) Prioritize the arrangement of the number of people to the large capacity laboratory, according to the laboratory priority from high to low laboratory arrangement.

Laboratory resources optimization arrangement function: f_2

3) Try not to arrange teachers to conduct laboratory courses consecutively in one day, and try to distribute the class time evenly, so as to reduce the teaching burden of teachers.

Teacher optimization function: f_3

Combined with the above function definition of the situation can be concluded that the soft constraints are relatively more complex, so the detailed definition of the objective function is placed in the back of the specific problem of research and implementation of the process of definition.

Experimental course examination arrangement problem is a multi-objective decision-making optimization problem under the constraints, according to the above objective function and constraints, the establishment of combinatorial optimization model is as follows:

$$\min f(x) = (f_1, f_2, f_3)$$

$$s.t.R: \begin{cases} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \end{cases} \tag{15}$$

2.2. Heuristic Algorithm for Optimization Solving

2.2.1. Genetic algorithms. Genetic Algorithm (GA) is a method for solving optimization problems based on a model of biological evolution [23].GA has several chromosomes, each of which can provide possible solutions to a given problem. The goal of a genetic algorithm is to evolve a population of candidate solutions (called phenotypes or individuals) to a given optimization problem. Genes and chromosomes are key concepts in genetic algorithms; genes that are on the same chromosome can be modified and mutated. In the traditional binary representation, solutions are represented using strings containing only 0s and 1s, but other encoding schemes can be used depending on the actual problem scenario.

The optimization process of a genetic algorithm usually involves the following steps:

1) Initialization: create a set of initial solutions, usually randomly generated.

2) Evaluation: Evaluate the quality of each solution using an objective function or other evaluation criteria.

3) Selection: Select a subset of the current solutions for subsequent operations, usually using roulette or tournament selection.

4) Crossover: Perform crossover operations on the selected solutions to generate new solutions.

5) Mutation: perform mutation operation on the new solution to avoid the solution set falling into local optimization and increase the diversity of the solution set at the same time.

6) Repeat: Repeat the execution of the second to the fifth step until the predetermined number of iterations or the solution objective is reached.

7) Output: output the found optimal solution.

Genetic algorithm, as a typical solution method for solving optimization problems, has a wide range of applications, such as engineering design, logistics planning, financial analysis, machine learning and other fields. It has the advantages of lower formalization and mathematical requirements of the problem and not easy to fall into the local optimal solution, but there are also practical problems such as higher arithmetic complexity and slow convergence speed, which need to be further improved to increase efficiency.

2.2.2. Population evolutionary strategy based on pyramid modeling. In traditional genetic algorithms, the initial feasible solutions are generated randomly. The advantage of this approach is that the generated solutions have more possibilities, but the disadvantage is that the quality of the generated solutions is not hierarchical. Meanwhile, in continuous problems, the randomly generated solutions are always within the feasible domain, so this approach can fully utilize its advantages, but in discrete problems, the randomly generated solutions have probability to be non-feasible solutions, so the arithmetic power will be wasted. The resource allocation optimization problem for university laboratories to be solved in this chapter belongs to the discrete-type problem, so the advantage of using traditional genetic algorithms to generate solutions randomly is more limited.

In the VPPM-VNS algorithm, the initial feasible solution is the set of all laboratory scheduling routes. From the perspective of genetics, the initial feasible solution represents the chromosome, and the laboratory equipment represents the gene. Therefore, the initial feasible solution is generated by first labeling the fitness of each gene according to the scheduling cost, then randomly selecting them using a roulette wheel method, and finally generating the genes in series. The fitness of each gene is calculated by the following equation:

$$r_k = \sum_{(i,j)}^{\Delta_i^+} \sum_{t=1}^{T} f_k * x_{ij}^t + Min \sum_{(i,j)}^{\Delta_i^+} \sum_{t=1}^{T} v_k * x_{ij}^t \tag{16}$$

$$r_{k\max} = MAX(r_k) \tag{17}$$

$$n_k = r_{k\max} - r_k \tag{18}$$

$$p_k = n_k / \Sigma n_k \tag{19}$$

where Equation (16) defines the formula for calculating equipment cost r_k. Equipment cost consists of two components: fixed cost and traveling cost. Equation (17) defines the formula for the calculation of $r_{k\max}$. Equation (18) defines the formula for calculating the actual value n_k of the equipment k when it participates in the roulette wheel calculation. Equation (19) p_k is the percentage of equipment k in the roulette wheel.

2.2.3. Variable population strategy based on pyramid modeling:

1) Parallel Evolution of Bottom Population. In order to accelerate the evolution of populations, the variable population evolution algorithm modeled as a pyramid introduces the idea of parallel evolution for populations that are at the bottom of the pyramid with a large population size. Figure 1 is a schematic illustration of population parallel evolution, showing the execution process of population parallel evolution and population confusion strategy. The initial population at the bottom of the pyramid is first divided into n sub-population and these sub-populations are allowed to evolve in parallel. Then the population confusion strategy, i.e., merging these sub-populations, is executed after a specified number of generations to maintain population diversity. Finally, the confused new population is re-divided into n sub-populations and these new sub-populations are allowed to continue to evolve in parallel, and this process is repeated until the upper level of the pyramid is reached.

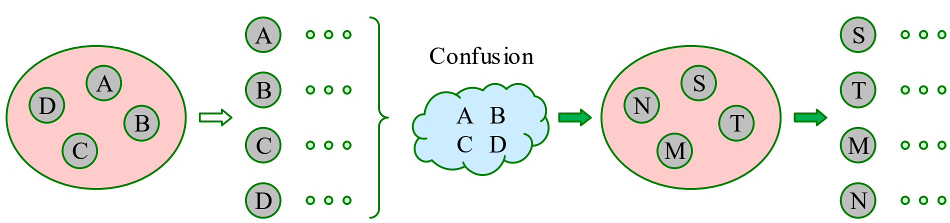

Fig. 1. Population parallel evolution

The implementation of the population confusion strategy is mainly accomplished by moving parallel evolved sub-populations to other sub-populations. The derivation process is as follows:

Let P be the initial population and the total number of individuals contained in the population is n:

$$P=\{x_1,x_2,x_3,x_4,x_5,x_6\dots,x_{n-1},x_n\} \tag{20}$$

Let W_1, W_2, W_3, W_4 be the four populations to be evolved in parallel, then:

$$W_i=\{x_{(n/4)\cdot(i-1)+1},\dots,x_{(n/4)\cdot i}\} \tag{21}$$

Let the number of individuals moving in the population be y, then the population after performing the population confusion operation is:

$$P=\{x_y,x_{y+1},\dots,x_{n-1},x_n,x_1,x_2,\dots,x_{y-2},x_{y-1}\} \tag{22}$$

The individuals within each subpopulation after the move are:

$$W_i=\{x_{(n/4)*(i-1)+y},x_{(n/4)+y+1},\dots,x_{(n/4)+2},\dots,x_{(n/4)+i+y-1}\} \tag{23}$$

The population confusion operation is now complete.

2) Population division. When the population has evolved to the generation where the pyramid level is located, the current population is divided into x group. If the population size is P and the total number of individuals contained in the population is n, then the value of P is:

$$P=\{x_1,x_2,x_3,\dots,x_{n-1},x_n\} \tag{24}$$

Let W_1, W_2, W_3, W_4, W_5 be the five populations to be divided, then the value of W_i is taken as:

$$W_i=\{x_{(n/5)+(i-1)+1},\dots,x_{(n/5)+i}\} \tag{25}$$

3) Population selectivity. Select a certain proportion of better adapted chromosomes from x group respectively.

Let the proportion of chromosomes to be eliminated from each population be v. The corresponding population screening strategy is:

Step 1: Sort each sub-population independently from smallest to largest scheduling cost:

$$W_1,W_2,W_3,W_4,W_5=Rank\{W_1,W_2,W_3,W_4,W_5\} \tag{26}$$

Step 2: Delete the inferior solutions in each population according to the elimination ratio v:

$$W_{new1}=(1-v)W_1,\dots,W_{new5}=(1-v)W_5 \tag{27}$$

4) Population merging. Merge the quality chromosomes selected from the x groups into a new population.

$$new\ P=\{W_{new1},W_{new2},W_{new3},W_{new4},W_{new5}\} \tag{28}$$

This completes a variant population operation. The variant population evolution process modeled on the pyramid structure is shown in Fig. 2, which demonstrates the core code of the pyramid variant population evolution operator, the 1st line of code is used to determine the relationship between the current number of iterations and the pyramid level, and the 2nd line of code starts to perform the variant population operation.

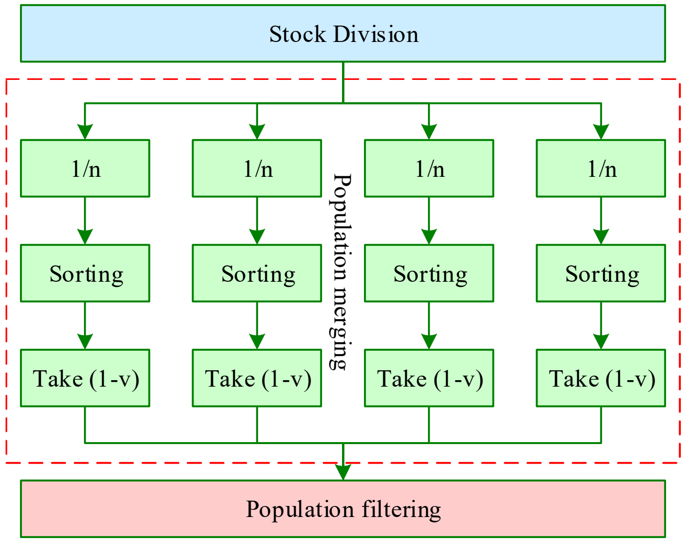

Fig. 2. The structure of the pyramid is motioned

2.3. Resource Scheduling Algorithm Design and Implementation

2.3.1. Scheduling algorithm framework. Figure 3 shows the framework of the scheduling algorithm proposed in this paper. The input to the program is a file containing information about nodes, Pods and their associated constraints. After obtaining the input information, then the resource occupation status of the current system is estimated, if the platform has sufficient lab resources free, then the tasks can be scheduled directly using the Kubernetes default scheduler, when there is fierce competition for resources, the platform will adjust the scheduling strategy to a batch scheduler or a dynamic scheduler by means of a manual or system monitor, and after the choice is made, then it will go to the corresponding scheduling process.

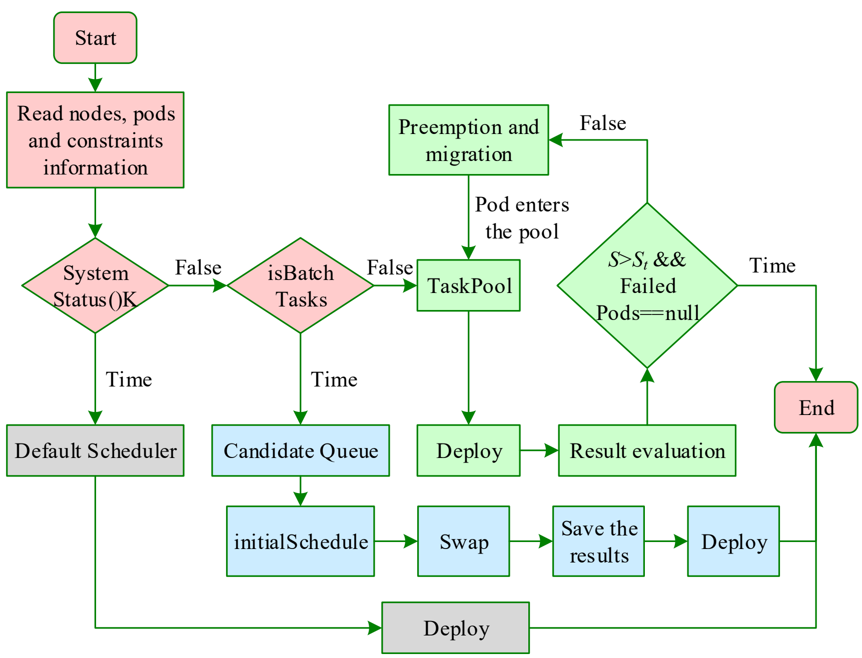

Fig. 3. The scheduling algorithm framework proposed in this paper

The batch scheduler is responsible for handling highly concurrent tasks, it will store all the tasks in a time period centrally in a candidate queue, all the tasks in the queue will be sorted according to a certain criterion, and then through the initialSchedule and Swap two steps to get the final deployment results.

The dynamic scheduler is responsible for handling tasks that contend for resources, in this paper we take GPU resources as an example. The dynamic scheduler selects tasks from TaskPool for deployment based on dynamic priority, and when there are tasks that cannot be scheduled and the current resource usage of the platform *S* does not reach the threshold *S*, the scheduler evaluates the platform to determine whether it needs to migrate the tasks.

2.3.2. Dynamic scheduler. The dynamic scheduler designed in this paper is a preemptive scheduling strategy that manages tasks competing for GPU resources in the lab. In the dynamic scheduler, TaskPool is its core data structure that stores the tasks that will be scheduled, which have dynamically changing priorities.

Equations (29)-(31) describe the methodology for calculating task priorities. When requesting the creation of a task, the algorithm in this paper requires the user to provide the value of resources Q_{res} required for the task to be deployed as well as the estimated runtime Q_{time}. Since this scheduling strategy is used within the laboratory, the data provided by the user is considered to be reliable in this paper, and the corresponding auditing procedures are also in place. For the required resource Q_{res}, it is calculated as in Equation (30), where W_i denotes the weight of the resource i, C_i denotes the number of resources i required for the task, and the resourcecelist includes scarce resources such as CPU and GPU. The estimation of time requires the user to first simulate the computation locally using a small dataset, which will be exemplified in the experimental part of this paper. Based on these two data, the default priority of each task is calculated using Equation (29) $P_{default}$. During the running of the task, the priority of the task P_{task} is set to be dynamically adjusted every ten minutes according to Equation (31). In Equation (31), W_t and R_t are the waiting time and running time of the task, respectively. Considering that the suspension and resumption costs of GPU tasks are very expensive, a task has to run for at least one hour before it can be preempted even if the priority of a task is lower than other tasks. The formula shows that when the waiting time of a task is longer, its priority is higher, and the running time is longer, its priority is lower.

$$P_{default} = \frac{1}{Q_{time}Q_{res}} \tag{29}$$

$$Q_{res} = \sum w_i * C_i, i \in resourcelist \tag{30}$$

$$P_{task} = P_{default} * \left(1 + \frac{w_t}{\max\left(R_t, 1\right)}\right) \tag{31}$$

The active tasks in the platform are stored in TaskPool and these tasks will be sorted according to priority from highest to lowest and the tasks in TaskPool are labeled as both scheduled and unscheduled. The dynamic scheduler will pause the scheduled but lower priority tasks and transfer the released resources to the unscheduled but higher priority tasks. Ideally, tasks can be deployed directly to those released resources, while in a few cases, the resource constraints of certain tasks cannot be met, in which case the algorithm calculates the resource usage of the platform and determines whether task migration is needed as a way to reduce resource fragmentation.

3. Simulation verification

3.1. Adaptation Simulation Results for Higher Education Laboratory Scheduling Programs

After modeling, simulation experiments will be conducted. The simulation will utilize genetic algorithm for resource scheduling to evaluate the overall performance of different scheduling loading schemes. The convergence and adaptability of the algorithm can be observed through several iterations of the simulation.

Simulation experiments are carried out on the model by setting the initialized number of populations as 20 and the number of genetic iterations as 100. Multiple simulation experiments are carried out to simulate the adaptability of the laboratory. Figure 4 shows the simulation results of the fitness and the number of treatments, Figure 4(a) shows the simulation results of the fitness of the scheduling scheme, and Figure 4(b) shows the simulation results of the number of integrated treatment devices loaded by the scheduling scheme.

From the simulation results, it can be seen that the scheduling strategy model based on genetic algorithm shows good convergence in the simulation process, such as Figure 4 (a), three experiments in the iteration number of 25 times, the adaptability value is gradually stabilized at 800. With the increase of the number of iterations, the adaptability gradually improves, and finally tends to be stabilized. This shows that the genetic algorithm can effectively search the solution space and find the near-optimal solution. After many simulation experiments, it can be seen that the genetic algorithm can find a solution close to the optimal solution, although it may not be guaranteed to find the global optimal solution, but under the given time and resource constraints, it can get a good approximate solution. Therefore, in the large-scale resource scheduling problem, using genetic algorithm model to evaluate the fitness and genetic operation of the laboratory resource scheduling scheme, a better solution can be found, and a near-optimal solution can be obtained by fast convergence.

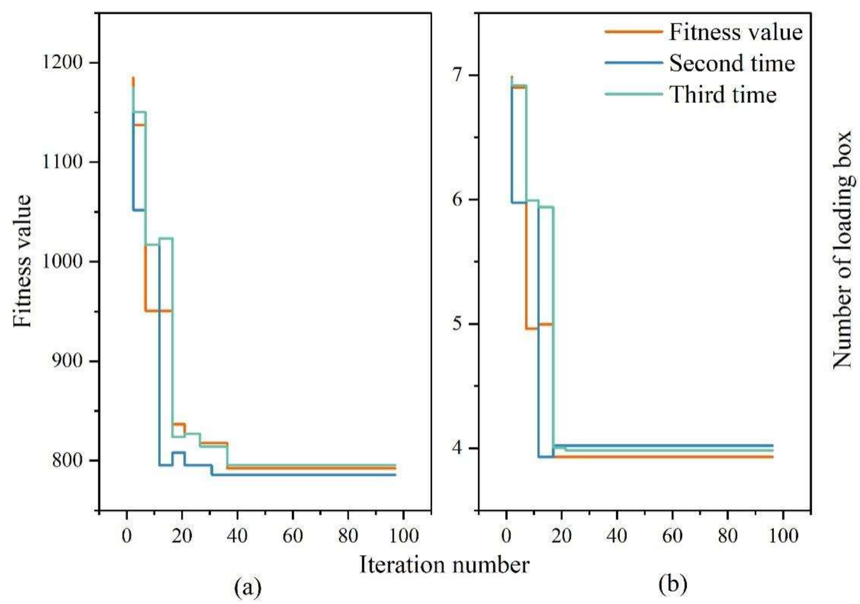

Fig. 4. The fitness and processing number simulation results

3.2. *Simulation results and time for different populations*

Simulation experiments are conducted again, the number of initialized populations is changed, and the algorithm simulation is conducted with 20 initialized populations, 100 initialized populations, and 200 initialized populations in turn, to analyze the effect of different initialized populations on the time of finding the near-optimal solution of the algorithm model and the time of generating the scheduling scheme, respectively, and the results of the experimental simulation are shown in Fig. 5, in which Fig. 5(a) is the result of the simulation with different populations, Fig. 5(b) is the time of simulation with different populations.) shows the experimental time of simulation for different populations.

From Fig. 5(a), it can be seen that the laboratory resource scheduling model based on genetic algorithm can get the near-optimal solution when the initialization populations are set to 20, 100 to 200, and when the number of simulation times is 10, the adaptations are 77.548, 77.916, 78.234, respectively. When the initialization populations are set to more digits, the model can get the near-optimal solution better. However, from Fig. 5(b), as the number of initialized populations increases, the time of simulation of this model increases, i.e., the time overhead of generating the scheduling scheme increases while better obtaining the near-optimal solution, and when the number of populations is 200, the time consumed is about 11s. Therefore, when using the genetic algorithm model to deal with the scheduling problem, for the actual parameter settings of the scheduling model, the parameter adjustment and algorithm optimization should be carried out according to the characteristics of the specific problem and the actual situation, in order to obtain a better performance and effect, and to improve the compliance of the scheduling model. At the same time, after generating the scheduling scheme, the whole scheduling scheme can be dynamically adjusted by combining the changes in the actual task volume and the uncertainty of the demand, so as to improve the applicability of the scheduling scheme through manual intervention.

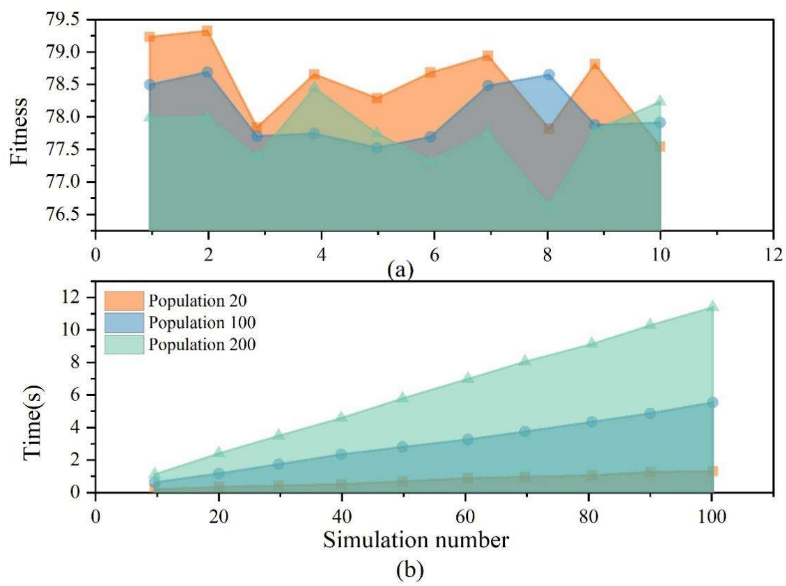

Fig. 5. Simulation results and time of different species

4. Analysis of examples

4.1. Distribution of laboratory equipment required by higher education institutions

All the solutions of the last generation of chromosome output of the improved genetic algorithm, the equipment scheduling of the laboratory courses of all classes and the scheduling of the laboratory courses are plotted by matlab, and the classification of equipment use according to the class can be seen that it can be a good scheduling of the use of the equipment, Fig. 6 shows the distribution of the amount of equipment required by each class at each time point, and the various scatter points in the figure indicate that, at a certain point in time, a certain class has the optimal scheduling strategy for laboratory equipment, for example, the coordinates of the example point in the figure are (19,9,1), indicating that the optimal amount of equipment to be scheduled for class 9's demand for the laboratory at the 19th time point is 1.

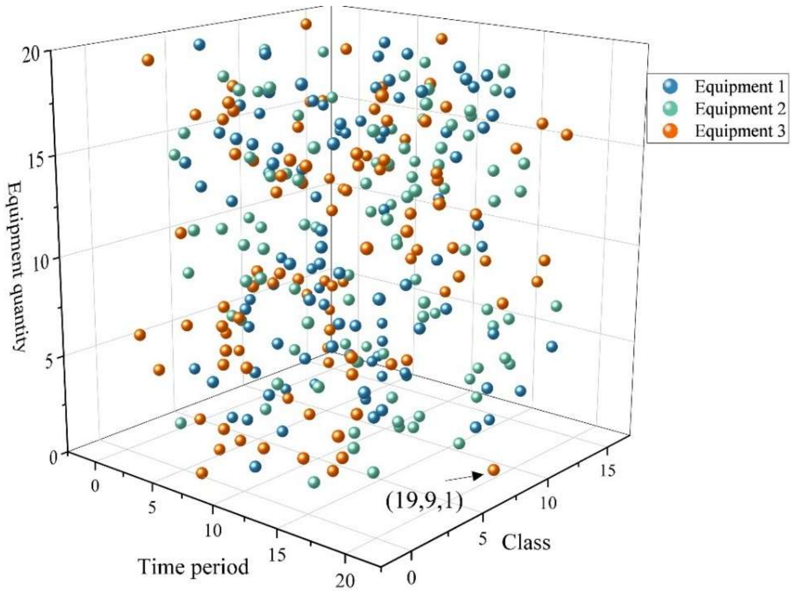

Fig. 6. The amount of equipment required for each class's time period

4.2. Analysis of practical applications

4.2.1. Class requirements. The statistics of all the solutions of the chromosome output of the last generation of the improved genetic algorithm are shown in Table 1, from which we know how each class has a demand for the laboratory and the laboratory equipment at a certain time point. From row 3, class 1 is at the 8th timestep, the 8th class to be prepared, the type of lab is 13th, and the number of equipment 1-3 demanded are 16, 14, and 13 respectively.

Table 1. Genetic algorithm optimal solution

Class	Time period	Course section	Experiment	Equipment 1	Equipment 2	Equipment 3
1	3	26	5	17	3	3
1	8	8	13	16	14	13
1	13	13	8	18	17	14
1	14	8	12	17	8	15
1	15	25	7	16	15	9
1	16	6	9	12	8	12
1	1	18	3	15	12	8
1	19	10	6	12	8	17
1	16	23	15	8	14	14
...	...	...	...	...	...	...
15	2	14	13	14	19	18
15	5	2	7	15	16	15
15	8	10	6	16	15	16
15	13	25	15	18	17	17
15	10	8	8	18	15	14
15	12	18	2	17	9	0
15	13	15	7	15	8	8
15	15	24	8	13	0	13
15	1	18	3	10	13	14
15	2	7	2	14	15	15
...	...	...	...	...	...	...

4.2.2. Laboratory equipment requirements. From the scheduling results, two classes a and b are randomly selected, and Figure 7 shows the demand for experimental equipment, (the course time is according to the “time period. Course period. Experiment type"). It can be seen that in the tenth week of the ninth session of the time point, the two classes at the same time to carry out experimental classes, respectively, experiment 4 and experiment 5. can be seen that the two demand for equipment resources are different, in which equipment 3 is a class focus on the need for the number of demand for the number of 17, while the b class for the equipment 3 has no demand, so the experiments when the equipment 3 can be all the scheduling allocation for the use of the a class, it can be seen that after the laboratory resources after intelligent scheduling, to achieve the educational resources, the lab can be used to provide the best possible results. It can be seen that after the intelligent scheduling of laboratory resources, the rational utilization of educational resources is realized, which is in line with the experimental expectations.

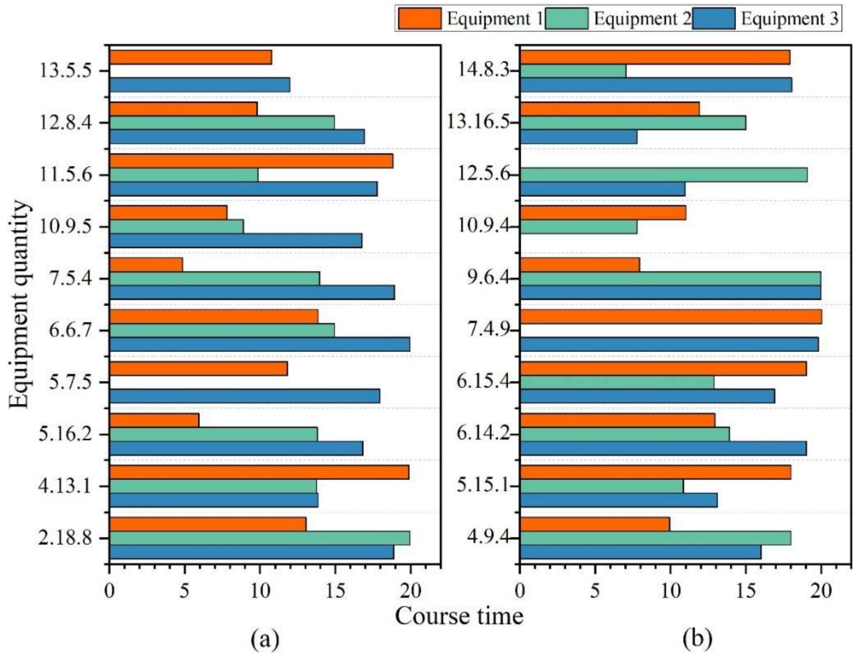

Fig. 7. Experimental equipment demand

4.2.3. Comparison of laboratory utilization before and after optimization. Figure 8 shows the comparison of equipment utilization before and after optimization, the colored columns in the figure are the highest utilization rate, and the white columns in the figure are the actual utilization rate, it can be seen that the improved genetic algorithm of the laboratory course resource scheduling has a significant increase in the utilization rate of the equipment relative to the traditional first-come-first-served equipment scheduling method, the utilization rate of the three kinds of laboratory resources and equipment is increased respectively by 16.3%, 34.6% and 18.4%, realizing the rational allocation of laboratory resources in universities.

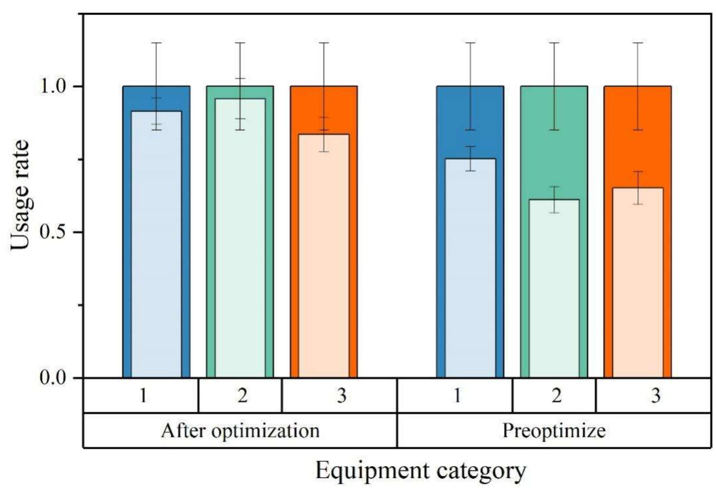

Fig. 8. The comparison of equipment usage before and after optimization

5. Conclusion

In this paper, for the task of resource scheduling in college laboratories, an optimization model is constructed, and the constraints of the model as well as the objective function are defined. And based on the genetic algorithm, design and realize the university laboratory resource equipment scheduling solution method. The effectiveness of the application of the model is verified through simulation experiments, and the scheduling strategy model based on genetic algorithm shows good convergence during simulation experiments, and the fitness value is gradually stabilized at about 800 when the number of iterations is 25 in all three experiments. The scheduling model can get the near-optimal solution when setting the initialized populations as 20, 100 and 200, and when the number of simulations is 10, the adaptations of these three population numbers are 77.548, 77.916 and 78.234, respectively. Through the arithmetic example to analyze the practical application effect of the model in the scheduling of laboratory resources in colleges and universities, and inputting the improved genetic algorithm into the auxiliary software, we get the experimental equipment scheduling of the course and the scheduling of the experimental course. After the improvement of university laboratory resource scheduling, the utilization rate of laboratory equipment is improved, and the utilization rates of the three kinds of laboratory resources and equipment are increased by 16.3%, 34.6% and 18.4% respectively.

References

[1] Bygstad, B., Øvrelid, E., Ludvigsen, S., & Dæhlen, M. (2022). From dual digitalization to digital learning space: Exploring the digital transformation of higher education. Computers & Education, 182, 104463.

[2] Ahel, O., & Lingenau, K. (2020). Opportunities and challenges of digitalization to improve access to education for sustainable development in higher education. Universities as Living Labs for Sustainable Development: Supporting the Implementation of the Sustainable Development Goals, 341-356.

[3] Horváth, L. (2023). Science-Industrial Experimental Model on Cloud Platform. In New Trends in Intelligent Software Methodologies, Tools and Techniques (pp. 76-84). IOS press.

[4] Holovnia, O. S., & Oleksiuk, V. (2022). Selecting cloud computing software for a virtual online laboratory supporting the Operating Systems course. In CTE 2021: 9th Workshop on Cloud Technologies in Education, December 17, 2021, Kryvyi Rih, Ukraine (No. 3085, pp. 216-227). CEUR Workshop Proceedings.

[5] Segrelles, J. D., Moltó, G., & Caballer, M. (2015, November). Remote computational labs for educational activities via a cloud computing platform. In 2015 Proceedings of the Information Systems Education Conference (ISECON) (pp. 309-321).

[6] Zhang, Y., Yao, J., & Guan, H. (2017). Intelligent cloud resource management with deep reinforcement learning. IEEE Cloud Computing, 4(6), 60-69.

[7] Alvarez, R., Mirzoev, T., Gowan, A., Henderson, B., & Kruck, S. E. (2019). Learning laboratories as services in private cloud deployment. Journal of Computer Information Systems, 59(4), 354-362.

[8] Xu, C., Wang, K., & Guo, M. (2017). Intelligent resource management in blockchain-based cloud datacenters. IEEE Cloud Computing, 4(6), 50-59.

[9] Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K., & Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Software: Practice and Experience, 47(9), 1275-1296.

[10] Gill, S. S., Chana, I., Singh, M., & Buyya, R. (2018). CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing. Cluster Computing, 21, 1203-1241.

[11] Yousafzai, A., Gani, A., Noor, R. M., Sookhak, M., Talebian, H., Shiraz, M., & Khan, M. K. (2017). Cloud resource allocation schemes: review, taxonomy, and opportunities. Knowledge and information systems, 50, 347-381.

[12] Ma, X., Wang, S., Zhang, S., Yang, P., Lin, C., & Shen, X. (2019). Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing. IEEE Transactions on Cloud Computing,

9(3), 968-980.

[13] Li, Y., Liu, J., Cao, B., & Wang, C. (2018). Joint optimization of radio and virtual machine resources with uncertain user demands in mobile cloud computing. IEEE Transactions on Multimedia, 20(9), 2427-2438.

[14] Calyam, P., Berryman, A., Welling, D., Mohan, S., Ramnath, R., & Ramnathan, J. (2014). VDPilot: feasibility study of hosting virtual desktops for classroom labs within a federated university system. International Journal of Cloud Computing 1, 3(2), 158-176.

[15] Arunarani, A. R., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. Future Generation Computer Systems, 91, 407-415.

[16] Domanal, S. G., Guddeti, R. M. R., & Buyya, R. (2017). A hybrid bio-inspired algorithm for scheduling and resource management in cloud environment. IEEE transactions on services computing, 13(1), 3-15.

[17] Strumberger, I., Bacanin, N., Tuba, M., & Tuba, E. (2019). Resource scheduling in cloud computing based on a hybridized whale optimization algorithm. Applied Sciences, 9(22), 4893.

[18] Arabnejad, V., Bubendorfer, K., & Ng, B. (2017). Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources. Future Generation Computer Systems, 75, 348-364.

[19] Gai, K., Qiu, M., Zhao, H., & Sun, X. (2017). Resource management in sustainable cyber-physical systems using heterogeneous cloud computing. IEEE Transactions on Sustainable Computing, 3(2), 60-72.

[20] Casas, I., Taheri, J., Ranjan, R., Wang, L., & Zomaya, A. Y. (2017). A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems. Future Generation Computer Systems, 74, 168-178.

[21] Chunting Xue,Feng Zhao,Xiangyong Chen & Jianlong Qiu. (2024). Adaptive neural preassigned-time tracking control for stochastic nonlinear systems with output constraint and its application. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering(7),1181-1192.

[22] Robert M Carrera,Chenxi Tao & Sunil K Agrawal. (2024). Designing and Analyzing In-Place Motor Tasks in Virtual Reality With Goal Functions. IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society

[23] Vasileios Charilogis & Ioannis G. Tsoulos. (2024). Introducing a Parallel Genetic Algorithm for Global Optimization Problems. AppliedMath(2),709-730.