Development of efficient testing algorithm for hydraulic valve group

Chang Liu1, *, Zhe Wang1, Keli Xing1, Xiajie Jin1
1School of Mechanical Engineering and Automation, Shanghai University, Shanghai, China
*Corresponding author: 704459798@qq.com

Abstract. Based on the principle of flexible testing technology, the test software is designed hierarchically and modularized. By calling the optimization algorithm of the test process, the automatic combination of the test process and the package call of the basic unit program of the test, the optimal sequencing and flexible configuration of the valve group test process are realized, and the efficiency of the automatic test scheme of the valve group is effectively improved. The test results show that the optimized test scheme can reduce the test time by 14.8% ~ 34.6% and the test energy consumption by 9.0% ~ 11.2%.

1. Introduction
Hydraulic valve manifold is usually a final product, which is produced through batch assembly. Due to the various types and complex structure of hydraulic valve manifold, the use of a highly efficient hydraulic test bench to test its quality is the key to the development and production of hydraulic valve manifold, and is also an important means to verify the performance index and reliability of the manifold. In this context, in order to meet the demand of factory testing of its high-volume and multi-type complex threaded cartridge valve manifold, this paper intends to develop a set of intelligent algorithms in the hydraulic valve manifold automatic testing system, and solve the problems of poor accuracy, low efficiency, long testing time and high energy consumption in the existing valve manifold testing system by combining intelligent algorithms to optimize the testing process, and meet the rapid and automatic testing of high-volume and multi-type valve manifolds.

2. Optimization of valve group test flow based on NSGA2 algorithm
The purpose of valve test flow optimization is to shorten the test time and reduce the test energy consumption, so the influencing factors are analyzed firstly.

(1) Module test sequence. The test of each module of the valve group has a sequence, that is, the test priority of the module. The module with higher priority must complete the test before the module with lower priority. Changing the test sequence between modules with the same priority does not affect the final test results.

(2) Number of module interfaces. There are a large number of electrical and hydraulic interface specifications required for valve block module testing, and the test bench usually cannot meet the interface requirements of valve block testing at one time. It is necessary to complete the whole valve block testing in stages according to the module testing sequence, and assign appropriate electrical and hydraulic interfaces to the tested valve block module.
(3) The module tests the flow and pressure. Because the flow and pressure of each module in the valve group are different, different system flow and pressure need to be adjusted when testing different valve group modules.

2.1. Optimization algorithm selection
Considering that the objective of the optimization problem of valve group test flow is test time and test energy consumption, there are only two optimization objectives, which belong to low dimensional multi-objective optimization problem, and there is no need to use nsga3 and rvea algorithms for high dimensional multi-objective optimization. SPEA2 pays too much attention to the global search ability, which leads to the lack of local search ability; Although MOEA / D is fast, it is only suitable for simple tasks with low dimension; NSGA2 has better comprehensive performance in low dimensional target, and the solution is more uniform [1]. To sum up, the NSGA2 algorithm is a suitable choice for the research content of this paper.

NSGA2 is improved from NSGA. The main difference between NSGA and the simple genetic algorithm is that the algorithm is stratified according to the dominance relationship before the selection operator is executed. Specifically, NSGA first ranks all individuals in the population in terms of non-dominance and assigns a virtual fitness to each stratum of non-dominated individuals, with the higher the non-dominance rank the greater the virtual fitness. Then a sharing radius is specified, and for a population individual, the sharing function of that individual and the rest of the individuals within the sharing radius is calculated, and the sharing function refers to the sum of the distances between that individual and the rest of the individuals, and subsequently the virtual fitness of that individual is divided by the sharing function to obtain the fitness of that individual, and finally the conventional genetic algorithm is operated according to the fitness.

Although NSGA has been applied in many problems, it has disadvantages such as high computational complexity, no elite strategy, and the need to specify the sharing radius. Compared with NSGA, NSGA2 has been improved for the disadvantages of NSGA by the following three aspects.

(1) A fast non dominated sorting method is proposed to reduce the computational complexity of the algorithm.

(2) The concept of crowding degree is proposed to replace the fitness sharing strategy which needs to specify the sharing radius, and it is used as the priority standard in the peer comparison after quick sorting to maintain the diversity of the population.

(3) The elite strategy is introduced to expand the sampling space. Combining the parent population with the offspring population to compete together to produce the next generation population is beneficial to keep good individuals into the next generation and improve the level of population evolution.

2.2. Construction of NSGA2 algorithm
The major difference between multi-objective optimization problems and single-objective optimization problems is that the solution of a multi-objective optimization problem should be a set, which is called the Pareto optimal solution, also called a non-inferior solution. In order to determine the Pareto optimal solution, the concept of dominance is used in the multi-objective optimization problem.

The purpose of multi-objective optimization algorithm is to generate new solutions through various methods to approximate the Pareto optimal solution. NSGA2 continues the idea of genetic algorithm, and introduces the concepts of Pareto optimal solution and crowding degree, which increases the selection pressure of individuals in the population, and speeds up the convergence speed of individuals to Pareto optimal solution on the basis of preserving the diversity of individuals. The algorithm flow is shown in Figure 1.
Fig. 1. Flow chart of NSGA2 algorithm.

NSGA2 algorithm actually belongs to a solution idea, its essence is to select individual population through Pareto optimization and crowding degree, while other conventional operations of genetic algorithm are determined by specific problems. Therefore, combined with the actual situation of valve group test process optimization problem, the following are several important parts of NSGA2 algorithm.

2.2.1. Population initialization
Population initialization is to generate an initial population. Because genetic algorithm uses chromosome to carry out various genetic operations, it needs to realize the mutual conversion of solution space and coding space through decoding matrix. The common arrangement methods include binary coding, indicator coding and permutation coding [2]. In this problem, the input object is module test sequence, which belongs to permutation type, so permutation coding is used.

2.2.2. Crossover operator
Crossover algorithm is the most effective global search link of genetic algorithm. Crossover is the operation of replacing and reorganizing the partial structure of two parent individuals to generate new individuals. Through crossover, the search ability of genetic algorithm is greatly improved. In this paper, partial matched crossover algorithm [3] is used.

2.2.3. Mutation operator
Mutation refers to the process of forming a new chromosome by changing a part of the elements in the chromosome. It can increase the diversity of the population and reduce the risk of the evolutionary algorithm falling into a local optimal solution. In this paper, chromosome fragments are used to reverse the mutation algorithm.

2.2.4. Penalty function constraint
In this problem, there are priority constraints between modules, so the order of modules with high priority must be before the module with low priority. However, due to the stochastic characteristics of genetic algorithm, it is difficult to deal with constraints when solving the problem with constraints, while penalty function can transform the constraint problem into unconstrained problem. In genetic algorithm, penalty function method needs to construct penalty constraint function first. If individual violates the constraint, it multiplies its target value by a penalty factor, which is usually a large number.

2.2.5. Fast non-dominated sorting
In order to classify the individual priority of the population, it is necessary to carry out the non-dominated sorting. In the non-dominated sorting, each individual in the population with the size of N needs to compare M objective functions with N-1 individuals in the population, and the complexity is \(O(MN)\). Each time of grading needs to carry out N comparisons, and the time complexity is \(O(MN^2)\). In the worst case, each Pareto level contains only one individual, so it needs to be graded n times, and the time complexity will rise to \(O(MN^3)\). It can be found that if the size of the population is large, it
needs a lot of comparative calculation to sort the population by using this method. In view of the defects of traditional non-dominated sorting, NSGA2 uses fast non-dominated sorting method to improve the calculation speed, and there are many fast non-dominated sorting methods. This paper uses the sequential search strategy algorithm \(^{(4)}\) (ENS-SS) as the fast non-dominated sorting method.

2.2.6. Congestion calculation
For any given population, the crowding degree of an individual in the population is defined as the average distance between the two closest points of the individual along each objective function in the target space. The specific calculation formula is as follows:

\[
i_d = \frac{1}{m} \sum_{j=1}^{m} |f_{j}^{i} - f_{j}^{i+1}|
\]

Among them, \(i_d\) is the calculated value of the crowding degree of the \(i\)-th member in the population; \(f_{j}^{i}\) is the \(j\)-th objective function value of the \(i\)-th member. Crowding degree describes the distribution of members near a certain member of the population. For individuals with high degree of crowding, the difference between nearby individuals is large; for individuals with low degree of crowding, the difference between nearby individuals is small. The degree of crowding essentially indicates the degree of diversification of individuals and also provides a basis for subsequent selection strategies.

2.2.7. Elite selection
The NSGA2 algorithm merges the parent individual \(P_t\) of the \(t\)-th generation with the offspring individual \(Q_t\) obtained through selection, crossover, and mutation to form a population \(R_t\) with a size of \(2N\), and then implements an elite selection strategy for the population \(R_t\). The elite selection strategy refers to sorting all individuals in the population \(R_t\), and the sorting follows the following rules.

1. First compare the non-dominated level, the individual with the higher non-dominated level wins.
2. If the level of non-dominance is the same, the individual with a high degree of congestion wins.
3. If the non-dominance level is the same and the crowding degree is the same, the individuals are randomly selected.

2.3. Algorithm implementation based on Python
As a general programming language that is closest to natural language\(^{(5)}\), using Python to write related algorithms can improve the efficiency of algorithm writing and debugging. Geatpy is a high-performance and practical evolutionary algorithm toolbox based on Python. It provides many evolutionary algorithm templates and various operation functions. Since there is no cumbersome packaging, you can use the functions provided by Geatpy to conveniently combine freely, realize and research. A variety of improved evolutionary algorithms solve problems that are difficult to solve by traditional optimization algorithms.

2.3.1. Parameter definition
First define the relevant parameters in the NSGA2 algorithm. Let the population size \(N=50\), the maximum genetic algebra \(\text{maxGen}=200\), the crossover probability \(P_c=1\), the mutation probability \(P_m=1\), and the penalty factor \(\sigma=10\). The optimization goal is the time and time of the test preparation process. Minimal energy consumption. Secondly, define the relevant input parameters and output parameters according to the actual situation.

2.3.2. Optimization results and analysis
Run the program and the results are shown below. Figure 2 shows the spatial distribution of the first, 50th, 100th and 200th generation solutions. Among them, the target 1 value and the target 2 value are the theoretically calculated values of the test preparation process time and energy consumption, respectively.
In general, with the increase of genetic algebra, the number of solutions that violate the rules gradually decreases until they disappear, the number of non-dominated solutions gradually increases, and the solution space gradually approaches to a specific shape, and the change is smaller and smaller. This shows that in this problem, after a long enough iterative evolution of NSGA2 algorithm, the spatial distribution of solutions tends to be stable.

However, the running time of the program is proportional to the genetic algebra. In the actual test, when the genetic algebra is 50 generations, the running time is 9.2 seconds; when the genetic algebra is 100 generations, the running time is 18.7 seconds, and when the genetic algebra is 200 generations, the running time is 36.5 seconds. So each generation of solution sets needs to be evaluated to select suitable genetic algebras.

3. Operation test and analysis of valve test system
In order to test the correctness of the optimization algorithm of valve group test flow designed in this paper, the original test scheme and three test schemes with different preferences (shortest time and Minimum energy consumption) are tested respectively, and the results are analyzed.

Under the three different schemes automatically generated by the test software, the number of test stages and the module test sequence of the tested valve group are shown in Table 1. The number in the table is the module serial number of the tested valve group, and the modules in each test stage are tested in sequence. After the test, the time power diagram of different test schemes obtained after data processing is shown in Figure 3.
Table 1. Test sequence of different schemes.

|                  | Original plan | Shortest time | Minimum energy consumption |
|------------------|---------------|---------------|---------------------------|
| Test phase 1     | 1,2           | 1,9,2,7       | 1,9,5,2                   |
| Test phase 2     | 3             | 8,10,4        | 8,10,4                    |
| Test phase 3     | 4,5,6,7       | 5,3,6         | 7,3,6                     |
| Test phase 4     | 8,9           |               |                           |
| Test phase 5     | 10            |               |                           |

According to the above test data, the actual total test time and the actual total test energy consumption can be obtained. Compared with the original test process, the test time can be reduced by 14.8% ~ 34.6%, and the test energy consumption can be reduced by 9.0% ~ 11.2%. It can be seen that no matter which process optimization scheme is selected, it is greatly improved compared with the original scheme.
4. Summary
A mathematical model is established to optimize the test time and energy consumption of valve group. NSGA2 algorithm is used for multi-objective optimization. The optimization program of valve group test process is written in Python. Two different preferred solutions, the shortest test time and the lowest test energy consumption, are obtained from the optimal solution set by cost performance method to meet the different optimization requirements of valve group test. Using the developed optimization algorithm, the factory performance of the tested valve group is tested. The test results show that compared with the original test scheme, the optimized test scheme can shorten the test time by 14.8%~34.6% and reduce the test energy consumption by 9.0%~11.2%.

References
[1] Xiao Xiaowei, Xiao Di, Lin Jinguo, Xiao Yufeng. Overview of multi-objective optimization [J]. Computer application research, 2011, 28 (03): 805-808 + 827
[2] Zhang Chaoqun, Zheng Jianguo, Qian Jie. Comparison of genetic algorithm coding schemes [J]. Computer application research, 2011, 28 (03): 819-822
[3] Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. Artificial Intelligence Review 13: 129–170, 1999.
[4] Zhang X, Tian Y, Cheng R, et al. An Efficient Approach to Non-dominated Sorting for Evolutionary Multi-objective Optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(2):201-213.
[5] Song Tian, Huang Tianyu, Li Xin. Python language: an ideal choice for teaching reform of programming course [J]. China University teaching, 2016 (02): 42-47