Hindawi
Complexity
Volume 2020, Article ID 7906019, 12 pages
https://doi.org/10.1155/2020/7906019

Research Article

A Test Procedure Optimization Method for an Industrial Robot Servo System on an Integrated Testing Platform

Shaomin Tang,[1] **Guixiong Liu**,[1] **Zhiyu Lin**,[1] **Xiaobing Li**,[2] **and Minqiang Pan**[1]

[1]*School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China*
[2]*CEPREI, Guangzhou 510610, China*

Correspondence should be addressed to Guixiong Liu; megxliu@scut.edu.cn and Minqiang Pan; mqpan@scut.edu.cn

Received 3 August 2020; Revised 18 September 2020; Accepted 19 October 2020; Published 21 November 2020

Academic Editor: Kang Li

A test procedure optimization method was proposed in this paper to improve the test efficiency of the industrial robot servo system (IRSS) to be tested on the IRSS integrated testing platform (IITP). First, an ordered sequence was used to define the IRSS test project when tested on the IITP. The ordered sequence consisted of execution subelements, which were a combination of control variable parameters of the IITP. Second, the optimization relationships among the IRSS test projects were dug out according to the IRSS test project sequences. Finally, by using the traveling salesmen problem (TSP), an optimization model was established based on the optimization relationships of the IRSS test projects, and the optimal test order of the IRSS test projects to be tested on the IITP was obtained by solving the model. A case analysis showed that the proposed method optimized the test procedure on the IITP effectively, and the test time when implementing the test on the IITP according to the optimized order was nearly 50% shorter than before optimization. The optimization effect was thus found to be significant.

1. Introduction

The industrial robot servo system (IRSS) is the execution unit that enables industrial robots to complete their assigned tasks and constitutes the core component of industrial robots [1]. The movement speed, positioning accuracy, bearing capacity, and operation performance of industrial robots are directly affected by the IRSS [2]. Moreover, accurate, rapid, and reliable testing of the performance of the IRSS is extremely important for improving the performance of industrial robots. In recent years, IRSS testing technology has gradually become mature, and it becomes a major trend to design an integrated testing platform for various testing projects. Zhu et al. [3] designed a test platform to accomplish the IRSS operating characteristics, including both steady state and dynamic state. Guo et al. [4] developed a platform to test the performance of IRSS accurately, reliably as well as conveniently. In addition, with the IRSS specific performance testing requirements, the functions of the test platform are further enriched. Yang et al. [5] designed dynamic tracking performance test on the test platform. Wang et al. [6] introduced an accelerated degradation test on the IRSS test platform. George et al. [7] designed and developed an automated test platform for IRSS at room temperature as well as at elevated temperatures. With the improvement of the integration of the IRSS test platform, problems related to the integration become more and more prominent. Sidhom et al. [8] proposed a higher-order sliding mode differentiator with dynamic gains to increase the test accuracy of the test platform. Zhang et al. [9] and Luo et al. [10] addressed to automation of the IRSS platform. Cheng et al. [11] propose a simulation platform to address the complex, costly, and tedious test process of servo systems.

In the previous research, we combed the existing IRSS test standards and technical requirements according to the test requirements of the IRSS and formed the IRSS test projects as shown in Figure 1. We then built an integrated testing platform on which to realize all the IRSS test projects, as shown in Figure 2. The IRSS integrated testing platform (IITP) can be used to conduct a comprehensive test for the IRSS and address the inefficiency caused by multiple projects being tested on different platforms. However, due to the

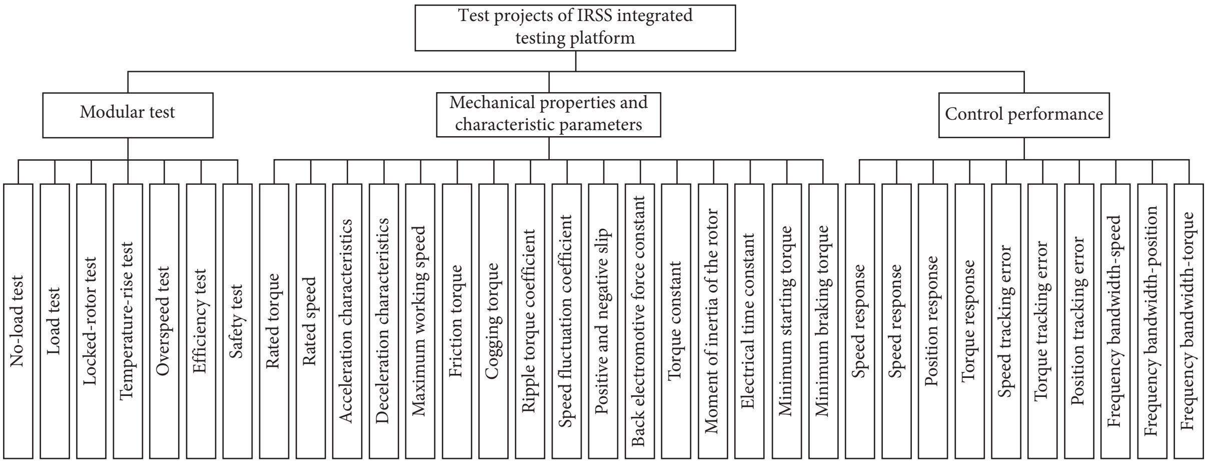


Figure 1: IRSS test projects of the IITP.

Figure 2: IRSS integrated testing platform (IITP).

large number of IRSS test projects being tested on the IITP, as well as the time-consuming nature of doing so, low testing efficiency remains an issue. Modular experimental designs [12] have been applied to parts of the IRSS test projects based on the IRSS test standard and technical requirement, as seen in Figure 1 as a modular test. For example, the no-load test of a modular test on the IITP can test four test projects: no-load current, no-load voltage, no-load power, and no-load speed. However, the efficiency remains nonideal, for there are still too many IRSS test projects that cannot be tested modularly.

We found that the test process of IRSS test projects on the IITP involves changing the control variables of the IITP and that some steps of the test process of IRSS test projects shared the same control variable parameters on the IITP. Based on this finding, we propose a test procedure optimization method for the IRSS being tested on the IITP. The novelty of the proposed method is that we find out a method to dig out the optimization relationships between IRSS test projects when tested on the IITP, and then we establish an optimization model based on this finding to get the optimal test order of the IRSS test projects to be tested on the IITP, which results in shortening the test time on the IITP.

The main contributions of this paper are the following:

(i) A method that uses an ordered sequence to define the IRSS test project is proposed according to the relationship between IRSS test projects and the control variable parameters on the IITP. The ordered sequence defining the IRSS test projects is formed by the execution subelements of IITP, which is the combination of IITP control variable parameters.

(ii) A method to shorten the test time by sharing the same IITP execution subelements among the IRSS test project sequences is discussed. The directed operation symbol "∩" is proposed to perform in two IRSS test project sequences while they were tested one after another. Two optimization relationships between IRSS test projects when tested on the IITP are identified by the "∩" operation, namely, the merger optimization relationship and the sequence optimization relationship.

(iii) The optimization problem of the test order of the IRSS test projects on the IITP is considered in the framework of the TSP, with a fixed starting point and ending point whose aim is to seek the shortest path through all cities. An optimization model is established, and the optimal test order with the shortest test time of the IRSS tested on the IITP is obtained by solving the model.

The remainder of this paper is organized as follows. In Section 2, an overview of the TSP and its applications and solutions is presented. In Section 3, the method of using an ordered sequence to define the IRSS test projects is proposed. In Section 4, the optimization relationships of the IRSS test projects based on the ordered sequence definition are presented. In Section 5, the IRSS test procedure optimization method, including the establishment of and solution to the model are introduced. In Section 6, an application case is used to verify the effectiveness of the proposed optimization method. In Section 7, the conclusion is presented.

2. An Overview of the TSP and Its Applications and Solutions

The TSP is based on a scenario including a salesman and n cities, in which the salesman must find the shortest path through all the cities [13]. Given a set of n cities $\{C_1, C_2, \ldots,$

$C_n\}$, with d_{ij} denoting the distance between city C_i and C_j, the TSP requires the Hamilton circle with the shortest distance through all the cities to be found. The decision variables are introduced as follows:

$$h_{ij} = \begin{cases} 1, & \text{if the salemen travel } C_j \text{ after } C_i, \\ 0, & \text{others.} \end{cases} \tag{1}$$

The objective functions can be derived as follows:

$$\min Z = \sum_{i,j=1}^{n} h_{ij} d_{ij}. \tag{2}$$

The TSP is a classical combinatorial optimization problem, which is often used as a reference to model practical problems. Yang et al. [14] formulated the path-planning problem of goods transportation as a TSP. Wei et al. [15] established the mathematical model of image-stitching based on the TSP. In addition to the basic model described above, there are many variants of the TSP, such as the TSP with drone [16], asymmetric TSP [17], multiple TSP [18], and colored TSP [19]. On the basis of the original model, some constraints were added to the variants to make them more suitable for modeling the practical problems. Shahmanzari et al. [20] referred to the election logistical problem as the roaming salesman problem, which can be characterized as a combination of the traditional periodic TSP and the prize-collecting TSP, with static arc costs and time-dependent node rewards. Aifadopoulou et al. [21] transformed the route optimization problem of the rebalancing vehicle into the one-commodity pickup and delivery capacitated TSP. Delle et al. [22] modeled the track routes for leaf collecting in the city as an asymmetric TSP, and by solving the model, 10–15% of the cost was cut down, with the consequence of reduced time, vehicle use, and fuel consumption.

The TSP and its variants are well-known NP-hard problems, whose difficulty to solve increases exponentially with the increase in the number of cities [23]. Solving the TSP and its variants has thus received attention from thousands since they were first proposed. The solutions to the TSP and its variants can be divided into deterministic and approximate solutions. Deterministic solutions to the TSP and its variants, such as cutting plane [24], branch and bound [25], and branch and cut [26] solutions, are unattractive due to the large amount of computation that they require. The approximate solutions to the TSP have attracted more attention due to their high solving efficiency and easy implementation. Intelligent search algorithms, one class of approximate solution to the TSP, have become a hotspot of research on TSP solutions in recent years as part of the growing interest in artificial intelligence. Kucukoglu et al. [27] introduced a new and effective hybrid simulated annealing/tabu search algorithm to solve the ETSPTW-MCR problem, and the proposed algorithm was capable of finding efficient and more realistic route plans for electric vehicles in computational studies. Liu et al. [28] proposed a niching particle swarm optimization algorithm based on Euclidean distance and hierarchical clustering for multimodal optimization and applied it to solve the TSP. Mitra et al. [29] reinterpreted the data compression problem as a TSP and introduced a modified ant colony algorithm in combination with a mutation operator to resolve the problem. Moreover, new intelligent search algorithms, such as the spider algorithm [30], chicken swarm optimization [31], and whale optimization algorithm [32], have also been used to solve the TSP and its variants.

3. An Ordered Sequence Description of IRSS Test Projects on the IITP

In our previous research, we found that the test of the IRSS test projects on the IITP are implemented by changing the parameters of the IITP control variables. The key IITP control variables relating to the IRSS testing are state of the IRSS being tested, state of the loading system, state running time, and acquisition of instrument/sensor signal. For example, when implementing load test, the operation process on IITP is as follows:

(i) The IITP starts the instrument/sensor to sample at a certain sampling frequency, controls the state of the IRSS being tested to reach its rated speed, and controls the state of the loading system to reach the rated load of the IRSS being tested.

(ii) The IITP then triggers the state running time when the state of the IRSS being tested and the state of the loading system are satisfied for testing; the timing is 30 min.

(iii) The corresponding instrument/sensor signal values are obtained by the time node of the completion of the state running time, and the performance indexes of the IRSS test projects are calculated in the background.

The process above shows that the test process of the load test can be defined by the combination of three control variables, including the state of the IRSS being tested, the state of the loading system, and the state running time. In order to determine the optimization relationship, we use code to define different parameters of the two control variables: the state of the IRSS being tested and the state of the loading system, respectively, as shown in Table 1. Hence, the test process of each IRSS test project can be defined by the combination of the parameter codes of the IITP control variable. In addition, some of the test process of the IRSS test projects have one step, and only one combination should be used to define this kind of IRSS test project, such as a no-load test is defined by (I_1, L_0, 30) and a load test by (I_1, L_1, 30). By contrast, some of the test process of the IRSS test projects have more than one step, and more than one combination should be used to define them; for example, the locked-rotor test is defined by (I_2, L_3, 5) (I_2, L_4, 1) orderly. Moreover, we take the combination of IITP control variable parameters as execution subelements of the IITP and use the ordered sequence of these subelements to define the test process of each IRSS test project on the IITP.

Table 1: The codes and definitions of the parameters of control variables on the IITP.

Control variables	The codes and definitions of the parameters of control variables					
State of IRSS being tested	I_0 Stop	I_1 Rated speech	I_2 Rated speech (reverse rotation)	I_3 Maximum allowable speed 120%	I_4 Linear increase to rated speed	I_5 Rated speech decreases linearly to 0
	I_6 Maximum working speed	I_7 Stop after rated speed	I_8 Linear increase to start-up	I_9 Step signal	I_{10} Chirp signal	I_{11} Sine wave frequency conversion signal
	I_{12} Linear increase to speed change of load system					
State of loading system	L_0 No-load output	L_1 Rated torque	L_2 Continuous locked torque	L_3 Peak locked torque	L_4 Linear load to rated torque	L_5 Linear increase to maximum torque

Let x_i be the subelement of the IITP, and it is the combination of three control variable parameters, that is, x_{si} representing the state of IRSS being tested, x_{li} representing the state of loading system, and x_{ti} representing the state running time. Let X be the set of m IITP execution subelements:

$$X = \{x_i | x_i = (x_{si}, x_{li}, x_{ti}), \quad i \in (1, m)\}, \tag{3}$$

where x_{si} and x_{li} are the code of the IITP control variable parameters as shown in Table 1; x_{ti}, which represents the state running time, is defined as the greatest common divisor of the time of the combinations that have the same x_{si} and x_{li}. That is, the combinations with the same x_{si} and x_{li} are treated as one IITP execution subelement, and x_{ti} of this element is the greatest common divisor of the time of all these combinations. For example, the efficiency test can be defined by the combination of (I_1, L_1, 60), and the speed fluctuation coefficient can be defined by (I_1, L_1, 10). Therefore, we can form an IITP execution subelement $x_j = (I_1, L_1, 10)$, where $x_{tj} = 10$, and the value 10 of x_{tj} is the greatest common divisor of 10 and 60. Consequently, the efficiency test consists of six IITP execution subelement x_j, and then it can be defined by the ordered sequence of $x_j x_j x_j x_j x_j x_j$. Similarly, the speed fluctuation coefficient consists of one IITP execution subelement x_j, and then it is defined by sequence of x_j.

Let Y be the set of n IRSS test projects. Each IRSS test project y_i can be described by the ordered sequence consisting of IITP execution subelements in the set X. Therefore, it can be derived from above that the test time of each IRSS test project can be determined by the state running time parameter of the subelements in the ordered sequence. Let N (y_i, x_j) represent the number of IITP execution subelements x_j in the ordered sequence y_i. We can thus obtain the test time of y_i by the following formula:

$$t(y_i) = \sum_{j=1}^{m} N(y_i, x_j) x_{tj}. \tag{4}$$

4. Optimization Relationships of the IRSS Test Projects on the IITP

Describing the IRSS projects with an ordered sequence of IITP execution subelements intuitively demonstrates that some of the IRSS test project sequences have the same IITP execution subelements. When testing multiple IRSS test projects on the IITP, if these same execution subelements of the IRSS test project sequences can be shared, the test time of the IRSS on the IITP can be optimized, which is the basis of the optimization method proposed in this paper. First, it must be decided whether the same execution subelements of the IRSS test project sequences can be shared when tested on the IITP. Since the IRSS test project sequence is ordered, the execution subelements of the sequence should be implemented on the IITP in a strict order determined by the sequence so as to fulfill the test process of the IRSS test project. Therefore, although the IRSS test projects have the same IITP execution subelements, those execution subelements may not necessarily be shared. Figure 3 shows an example for those who have the same execution subelements in the IRSS test project sequences, but the same execution subelements cannot be shared in the test procedure on the IITP. The IRSS test projects in Figure 3 have the same execution subelement x_2, but to test y_i, the execution subelement x_3 should be implemented after x_2 to complete the test of y_i, and no other execution subelement can be inserted in the middle of the test sequence of y_i. This is similar for the testing of y_j. Therefore, the IRSS test project y_i and y_j cannot share the execution subelement x_2, even though it is the same execution subelement in their test project sequences.

Based on the analysis above, we introduce a directed operation symbol "∩," by which $y_i \cap y_j$ means searching for the same execution subelement sequence as the sequence of y_i from the beginning of the sequence of y_j, such that the end of the sequence must cover the last execution subelement of y_i or y_j; otherwise, $y_i \cap y_j = \varnothing$. The ordered sequence obtained from the operation "∩" of the two test project sequences is that whose execution subelements can be shared in the test procedure of these two test projects tested on the IITP. Two optimization relationships between the IRSS test project

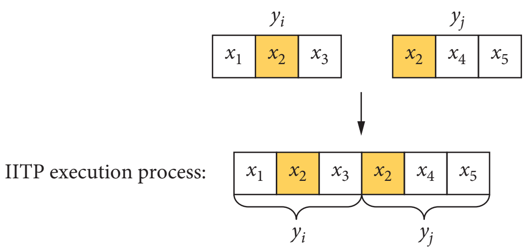


Figure 3: Schematic diagram of two IRSS test projects that cannot share a test process on the IITP, even though they have the same execution subelement in their test sequences.

sequences are determined according to the operation "∩": the merger optimization relationship and the sequence optimization relationship, respectively, as shown in Figure 4.

4.1. The Merger Optimization Relationship. The merger optimization relationship means that one IRSS test project sequence contains the sequence of another IRSS test project. As is shown in Figure 4(a), the IRSS test project sequence y_k contains the sequence of y_l. Let y_k be the containing test project and y_l be the contained test project. The calculation result of "∩" of y_k and y_l is thus as follows:

$$\begin{aligned} y_k \cap y_l &= x_2 x_2 x_3 = y_l; \\ y_l \cap y_k &= \varnothing. \end{aligned} \tag{5}$$

As can be seen from formula (5), when two IRSS test project sequences y_i and y_j have the calculation result of $y_i \cap y_j = y_j$, we say that y_i and y_j have the merger optimization relationship and that y_i is the containing test project, while y_j is the contained test project. When two IRSS test projects have the merger optimization relationship, the performance indexes of these two test projects can be obtained from one containing test project implemented on the IITP. Hence, the test time of these two test projects on the IITP can be shortened to that of the containing test project.

4.2. The Sequence Optimization Relationship. The sequence optimization relationship means that there are the same sequences between the end of one IRSS test project sequence and the front of another IRSS test project sequence. As shown in Figure 4(b), the execution subelement sequence $x_3 x_4$ at the end of IRSS test project sequence y_g is the same as that at the front of y_h. Let y_g be the front test project and y_h be the end test project. The calculation result of "∩" of y_g and y_h is thus as follows:

$$\begin{aligned} y_g \cap y_h &= x_3 x_4; \\ y_h \cap y_g &= \varnothing. \end{aligned} \tag{6}$$

As can be seen from formula (5), when two IRSS test project sequences y_i and y_j have the calculation result of $y_i \cap y_j \neq \varnothing$, we say that y_i and y_j have the sequence optimization relationship and that y_i is the front test project, while y_j is the end test project. When two IRSS test project sequences have the sequence optimization relationship, the execution subelement sequence obtained by the calculation of "∩" of these two test project sequences can be shared when implementing the test of the front test project first and then the end test project on the IITP. Therefore, the test time of these two IRSS test projects on the IITP can be reduced by the test time of that of the execution subelement sequence they shared.

5. IRSS Test Procedure Optimization Method on the IITP

The IRSS test procedure optimization method on the IITP proposed in this paper is to establish an optimization model based on the optimization relationships of the IRSS test project sequences introduced above and to obtain an optimal test order of the IRSS test projects to be tested on the IITP by solving the model.

5.1. The Establishment of the Optimization Model of IRSS Test Procedure on the IITP. For an IRSS with n test projects to be tested on IITP, there are $n!$ test orders. In addition, according to the optimization relationships derived from last section, for a certain test order L, the test time is calculated as follows:

$$t(L) = \sum_{i=1}^{n} t(y_i) - \sum_{i=1}^{n-1} t[L(i) \cap L(i+1)], \tag{7}$$

where $L(i)$ represents the IRSS test project ranking i in the test order L. As can be seen from formula (7), when the test order is determined, the test time on the IITP is the total test time of all the IRSS test projects minus the sum of the optimized test time of every two ordered test projects in the test order. Therefore, the aim of the optimization of IRSS test procedure on the IITP is to find a test order to obtain the shortest test time, that is, to minimize formula (7).

In order to further explore the relationship between the test order and the test time, a sequential test time expression is defined to express the test time of the test project y_j after the execution of another test project y_i, as follows:

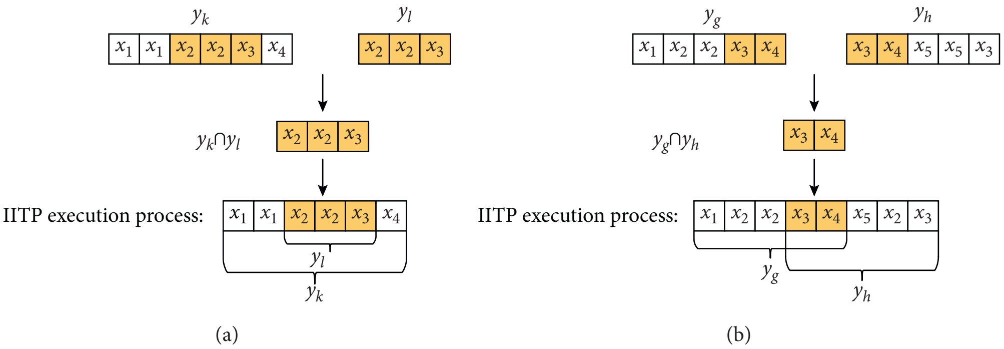
$$s_{ij} = t(y_i \longrightarrow y_j) = t(y_j) - t(y_i \cap y_j). \tag{8}$$

As can be seen from formula (8), when testing y_j after y_i on the IITP, the test time of y_j is the original test time of y_j minus the test time of the sharing execution subelement sequence of the two test projects.

Here, a virtual test project y_0 is introduced as the starting and ending test project of the test procedure; that is, it ranks first and last in the test order. The test project sequence of y_0 is empty. Hence, according to the definition of operation "∩," for every arbitrary test project sequence y_k, there are expressions as follows:

$$\begin{aligned} t(y_0) &= 0; \\ t(y_0 \cap y_k) &= t(y_k \cap y_0) = 0. \end{aligned} \tag{9}$$

In addition, according to formula (8), the following expressions can be obtained:



y_k
y_l
x_1 x_1 x_2 x_2 x_3 x_4
x_2 x_2 x_3
$y_k \cap y_l$
IITP execution process:
(a)
y_g
y_h
x_1 x_2 x_2 x_3 x_4
x_3 x_4 x_5 x_5 x_3
$y_g \cap y_h$
x_3 x_4
x_1 x_2 x_2 x_3 x_4 x_5 x_2 x_3
(b)

Figure 4: Two optimization relationships between the IRSS test project sequences: (a) the merger optimization relationship; (b) the sequence optimization relationship.

$$t\left(y_0 \longrightarrow y_k\right) = t\left(y_k\right); \qquad t\left(y_k \longrightarrow y_0\right) = 0. \tag{10}$$

Then, introducing the virtual starting and ending test project y_0 to the test order L which consists of n IRSS test projects, and combining formulas (8) and (10), the test time of the IRSS that is being tested in the test order L on the IITP can be updated as follows:

$$t(L) = \sum_{i=0}^{n} t\left[L(i) \longrightarrow L(i+1)\right] = \sum_{i=0}^{n} s_{i,i+1}. \tag{11}$$

As seen from formula (11), when the test order is determined, the test time is the sum of the test time of every two ordered test projects in the test order being tested sequentially. If taking each IRSS test project as a city and the test time of every two test projects being tested sequentially on the IITP as the path cost between two cities, the optimization problem of IRSS procedure on the IITP can be transformed into the TSP, with fixed starting and ending points, whose aim is to seek the shortest path traversing all the cities. Figure 5 shows the corresponding relation between the TSP and the optimization problem of IRSS test procedure on the IITP.

After relating the TSP and the optimization problem of IRSS test procedure on the IITP, the optimization model of IRSS test procedure on the IITP can be established according to the objective function of the TSP which formulated as (1) and (2). Then, we get the model expressed as follows:

$$\min t(L) = \sum_{i,j=1}^{n} x_{ij} s_{ij}. \tag{12}$$

$$\text{S.T}\begin{cases} \sum_{i=1}^{n} x_{ij} = 1; \quad j = 1, 2, \ldots, n, \\ \sum_{j=1}^{n} x_{ij} = 1; \quad i = 1, 2, \ldots, n, \\ x_{ij} = \{0, 1\}; \\ i = 1, 2, \ldots, n; \\ j = 1, 2, \ldots, n, \end{cases} \tag{13}$$

where formula (12) means that the aim of the optimization problem of IRSS test procedure on the IITP is to find a test order, according to which we tested on the IITP resulting in the shortest test time. Formula (13) means that every IRSS test project should be tested on the IITP and tested only once. In formula (13), x_{ij} is the decision variables, whose value is 0 or 1, and its expression is as follows:

$$x_{ij} = \begin{cases} 1, & y_i \longrightarrow y_j, \\ 0, & \text{else}. \end{cases} \tag{14}$$

5.2. The Solution to the Optimization Model of IRSS Test Procedure on the IITP. As mentioned above, there are $n!$ test orders with n IRSS test projects to be tested on the IITP; that is, the solving scale of the optimization model of the IRSS test procedure on the IITP will increase exponentially with the number of the IRSS projects to be tested. Similar to the TSP, it is also an NP-hard problem. We use the simulated annealing (SA) algorithm [33] to solve the optimization model in this paper. The pseudocode of SA algorithm is as follows (Algorithm 1):

The pseudocode of the SA algorithm shows that the algorithm includes two cycles: an inner cycle and an outer cycle. The inner cycle accepts the new solution according to the metropolis probability criterion, as a result of that even the suboptimal solution is likely to be accepted, thus effectively breaking out of the local optimal solution. Moreover, the outer cycle simulates the annealing process, with which the probability of accepting a suboptimal solution decreases, meaning that the solution obtained in each annealing process can converge, leading to the optimal solution. The inner and outer cycle solution mechanisms ensure that the solution obtained by the SA algorithm has a high probability of being the global optimization.

There are three key design points when using the SA algorithm to solve the optimization model of IRSS test procedure on the IITP: the generation of an initial solution and a new solution, the determination of the fitness of the solution, and the acceptance rules of the new solution. These three key design points are introduced as follows:

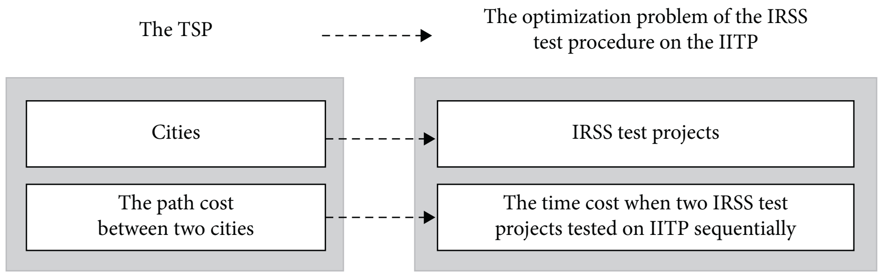

The TSP
The optimization problem of the IRSS test procedure on the IITP
Cities
IRSS test projects
The path cost between two cities
The time cost when two IRSS test projects tested on IITP sequentially

Figure 5: Corresponding relation between the TSP and the optimization problem of IRSS test procedure on the IITP.

(i) Generation of an initial solution and a new solution: the solution of the optimization model of IRSS test procedure on the IITP is the test order of IRSS test projects to be tested on the IITP. The SA algorithm is not sensitive to the initial solution, that is, whether the optimal solution obtained or not is not related to the initial solution. Therefore, when solving the optimization model, the initial solution L_0 is thus the input order of IRSS test projects to be tested. Moreover, the generation of a new solution L_{new} is to randomly select two test projects in the current solution L_{current} and to exchange their positions in L_{current}. Figure 6 shows the generation process of new solution L_{new}.

(ii) Determination of the fitness of solution: the fitness of solution refers to whether the solution is good or bad, which is judged in the solving process of the optimization model of IRSS test procedure on the IITP by whether the test time obtained by testing according to test order of the solution is short or long. The shorter test time the solution gets, the better the solution is. Therefore, the fitness of the solution can be determined by formula (11).

(iii) Acceptance rules of the new solution: when the new solution L_{new} generated, it is accepted according to the metropolis probability criterion. The specific steps are as follows:

(a) If $t\ (L_{\text{new}}) - t\ (L_{\text{current}}) < 0$, that is, the new solution is better than the current one, the new solution L_{new} is accepted, and let $L_{\text{current}} = L_{\text{new}}$.

(b) If $t\ (L_{\text{new}}) - t\ (L_{\text{current}}) \geq 0$, that is, the new solution is worse than the current one, then calculate the state transition probability by the following formula:

$$P_T\left(L_{\text{current}} \longrightarrow L_{\text{new}}\right) = \exp\left(\frac{-\left(t\left(L_{\text{new}}\right) - t\left(L_{\text{current}}\right)\right)}{T_k}\right). \tag{15}$$

(c) Randomly select a number r from a uniformly distributed interval of $(0, 1)$. If $P_T < r$, the new solution L_{new} is accepted, and let $L_{\text{current}} = L_{\text{new}}$. Otherwise, the new solution is discarded, and the original current solution is retained for the next iteration.

The solution of the optimization model of IRSS test procedure on the IITP based on the SA algorithm can be designed according to the three key design points introduced above, and the optimal test order for IRSS test projects to be tested on the IITP would be obtained by solving the optimization model. Figure 7 shows the flowchart of solving the optimization model of IRSS test procedure on the IITP based on the SA algorithm.

6. Case Analysis

In this section, the method proposed in the paper is applied to test procedure optimization of an IRSS in the development phase on the IITP. There are three IRSSs being tested, each with the same test projects, namely, a ripple torque coefficient, a speed fluctuation coefficient, a positive and negative slip, an electrical time constant, a safety test, a no-load test, a load test, a locked-rotor test, a temperature-rise test, an overspeed test, and an efficiency test, with total of 11 IRSS test projects. All the IRSS test projects above can be tested on the IITP.

First, seven execution subelements are constructed according to the relationship between the IRSS test projects to be tested and the control variables on the IITP as follows:

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} I_0 & L_0 & 30\,\text{min} \\ I_1 & L_0 & 5\,\text{min} \\ I_2 & L_0 & 5\,\text{min} \\ I_3 & L_0 & 2\,\text{min} \\ I_1 & L_1 & 10\,\text{min} \\ I_1 & L_2 & 5\,\text{min} \\ I_1 & L_3 & 1\,\text{min} \end{pmatrix}. \tag{16}$$

Second, the ordered sequence of the IRSS test projects are defined according to the seven execution subelements, as shown in Table 2. In Table 2, if each IRSS test project is tested without optimizing the test procedure on the IITP, the test time of one IRSS is 273 min, and the total test time of the three IRSSs is 819 min. The test time is thus too long to meet the urgent testing needs in the development phase.

Finally, the SA algorithm is adopted to solve the optimization model. The initialization parameters of the SA

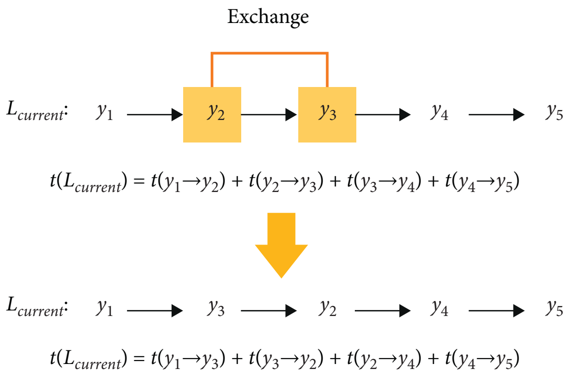

Exchange
$L_{current}$: y_1 y_2 y_3 y_4 y_5
$t(L_{current}) = t(y_1 \rightarrow y_2) + t(y_2 \rightarrow y_3) + t(y_3 \rightarrow y_4) + t(y_4 \rightarrow y_5)$
$L_{current}$: y_1 y_3 y_2 y_4 y_5
$t(L_{current}) = t(y_1 \rightarrow y_3) + t(y_3 \rightarrow y_2) + t(y_2 \rightarrow y_4) + t(y_4 \rightarrow y_5)$

Figure 6: Schematic diagram of the generation process of new solution L_{new}.

algorithm are as follows: initial temperature $T_k = 100$, times of internal cycles res = 20, temperature drop coefficient lam = 0.95, and cutoff temperature $T_f = 10$. Table 3 shows the optimizing schedule of the optimization model of application case based on the SA algorithm. When using the SA algorithm to solve the optimization model of IRSS test procedure on the IITP, it is with high solving efficiency that the optimal solution is obtained in the 20th iteration. In the 10th iteration and the 15th iteration, the current optimal solution was 148 min, but by the 20th iteration, the SA algorithm successfully jumps out from the local optimal solution and finally obtains the global optimal solution of 138 min, proving the effectiveness of the SA algorithm. Figure 8 shows the schematic diagram of the optimal test procedure on the IITP of application case. The optimal test order for the 11 IRSS test projects on the IITP is $y_5 \longrightarrow y_9 \longrightarrow y_{11} \longrightarrow y_7 \longrightarrow y_1 \longrightarrow y_4 \longrightarrow y_8 \longrightarrow y_{10} \longrightarrow y_6 \longrightarrow y_2 \longrightarrow y_3$, and the total test time is 138 min, which is only 50.55% of the test time without optimization. When testing the three IRSSs, the total test time is shortened from the original 819 min to 414 min, and the total optimization time is 405 min, showing a significant optimization effect of the proposed method. Therefore, the IRSS test procedure optimization method proposed in this paper can effectively shorten the test time of the IRSS on the IITP and improve the testing efficiency of the IRSS significantly.

In fact, when the IRSS test project is described by an ordered sequence of IITP execution subelements, parallel testing can be realized on the IITP. The purpose of parallel testing is to improve test efficiency by integrating the test of test projects according to their test resource [34]. In order to further demonstrate the superiority of our proposed method, parallel testing on the IITP is designed according to the IRSS test project sequences. Figure 9 shows the parallel testing result of the application case. As shown in Figure 9, the test time of the IRSS of application case on the IITP based on parallel testing is equivalent to the test time of sequential test of the IRSS test project y_9, y_6, y_{10}, y_4, y_8, y_3, and y_5, that is, 163 min. As derived from above, the test time of application case on the IITP without optimization is 273 min. Therefore, the test time of application case based on parallel testing is 110 min shorter than that without optimization, and the optimization ratio of parallel testing is 40.29%. However, comparing to our method, the test time of the application case based on parallel testing is 25 min longer than that of our method, with 10% lower in the optimization ratio. The reason of this phenomenon is that the parallel testing only can deal with the merger optimization relationship between the IRSS test project sequences but cannot deal with the sequence optimization relationship, resulting in the failure to utilize the sequence optimization in the test process.

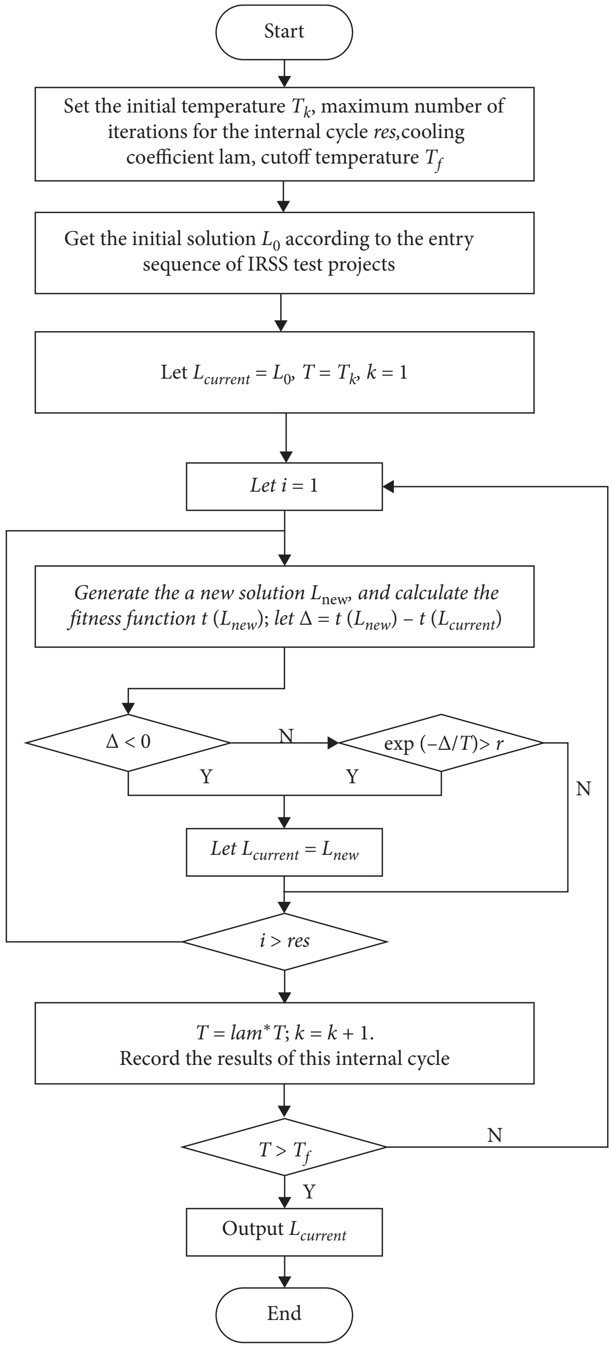

Start
Set the initial temperature T_k, maximum number of iterations for the internal cycle *res*,cooling coefficient lam, cutoff temperature T_f
Get the initial solution L_0 according to the entry sequence of IRSS test projects
Let $L_{current} = L_0$, $T = T_k$, $k = 1$
Let $i = 1$
Generate the a new solution L_{new}, and calculate the fitness function $t(L_{new})$; let $\Delta = t(L_{new}) - t(L_{current})$
$\Delta < 0$
N
$\exp(-\Delta/T) > r$
Y
Y
N
Let $L_{current} = L_{new}$
$i > res$
$T = lam^{*}T$; $k = k + 1$.
Record the results of this internal cycle
$T > T_f$
N
Y
Output $L_{current}$
End

Figure 7: Flowchart of solving the optimization model of IRSS test procedure on the IITP based on the SA algorithm.

In addition, we conduct a comparison experiment based on the solving method to the optimization model of IRSS test

```
Initialized ($T_k$, $T_f$, lam, res, $S_0$);
Let $T = T_k$, $S_{\text{current}} = S_0$
If $T > T_f$
Repeat
  For $k = 1$ to res do
  Begin
    Generate ($S_{\text{new}}$ from $C$)
    If $f(S_{\text{new}}) < f(S_{\text{current}})$, Then $S_{\text{current}} = S_{\text{new}}$
      Else if $\exp\{-[f(S_{\text{new}}) - f(S_{\text{current}})]/T_k\} > r$, Then $S_{\text{current}} = S_{\text{new}}$
    End
  $k = k + 1$
  $T = \text{lam} * T$
  End
End
```

Algorithm 1: Pseudocode for the simulated annealing algorithm.

Table 2: The ordered sequence and the test time of the IRSS test projects of application case.

ID	IRSS test projects	Code of IRSS test projects	IRSS test project sequences	Test time (min)
1	Ripple torque coefficient	y_1	x_5	10
2	Speed fluctuation coefficient	y_2	x_2 x_2	10
3	Positive and negative slip	y_3	x_2 x_3	10
4	Electrical time constant	y_4	x_6 x_6	10
5	Safety test	y_5	x_1	30
6	No-load test	y_6	x_2 x_2 x_2 x_2 x_2 x_2	30
7	Load test	y_7	x_5 x_5 x_5	30
8	Locked-rotor test	y_8	x_6 x_7	6
9	Temperature-rise test	y_9	x_5 x_5 x_5 x_5 x_5 x_5	60
10	Overspeed test	y_{10}	x_4 x_2 x_2 x_2	17
11	Efficiency test	y_{11}	x_5 x_5 x_5 x_5 x_5 x_5	60

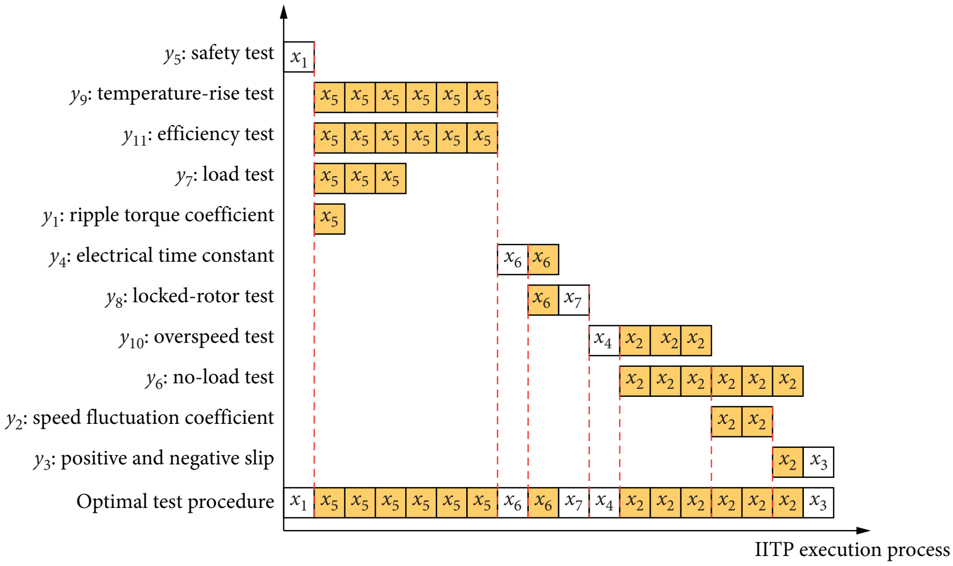

y_5: safety test
x_1
y_9: temperature-rise test
x_5 x_5 x_5 x_5 x_5 x_5
y_{11}: efficiency test
x_5 x_5 x_5 x_5 x_5 x_5
y_7: load test
x_5 x_5 x_5
y_1: ripple torque coefficient
x_5
y_4: electrical time constant
x_6 x_6
y_8: locked-rotor test
x_6 x_7
y_{10}: overspeed test
x_4 x_2 x_2 x_2
y_6: no-load test
x_2 x_2 x_2 x_2 x_2 x_2
y_2: speed fluctuation coefficient
x_2 x_2
y_3: positive and negative slip
x_2 x_3
Optimal test procedure
x_1 x_5 x_5 x_5 x_5 x_5 x_5 x_6 x_6 x_7 x_4 x_2 x_2 x_2 x_2 x_2 x_2 x_3
IITP execution process

Figure 8: Schematic diagram of the optimal test procedure of application case.

procedure on the IITP. We use the genetic algorithm (GA) [35] to solve the optimization model of application case, and the solutions obtained from the GA with three different combinations of initialization parameters are compared and analyzed with that of the SA algorithm with the initialization parameters the same as above. Table 4 shows the initialization parameters of the GA and the SA algorithms in the comparison experiment. In the three combinations of initialization parameters of the GA, the crossover probability P_c and mutation probability P_m are the same, and the

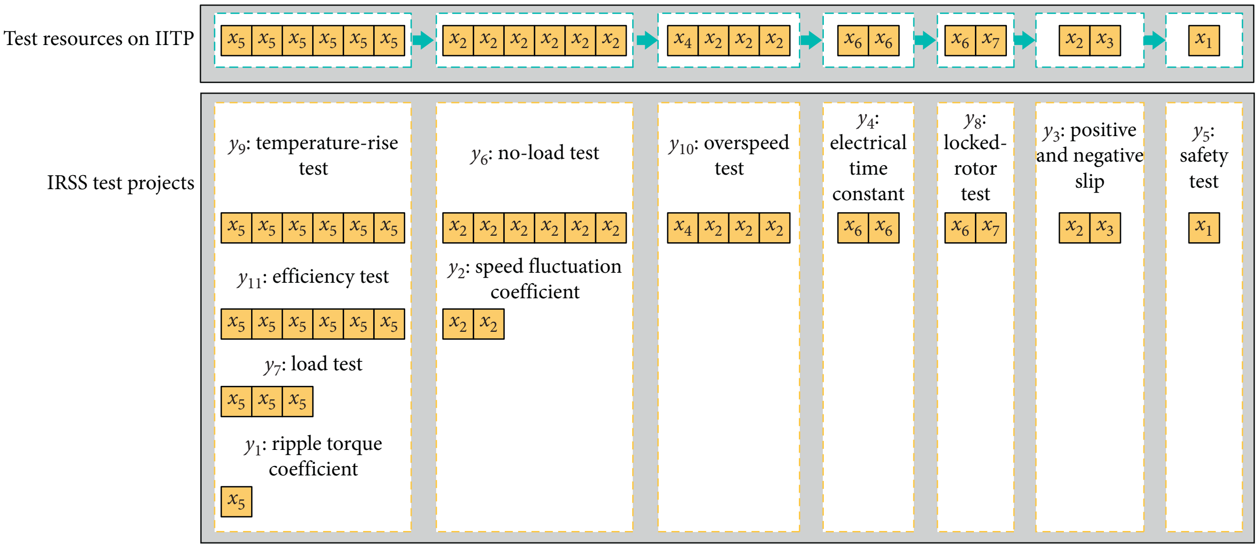

Test resources on IITP
x_5 x_5 x_5 x_5 x_5 x_5
x_2 x_2 x_2 x_2 x_2 x_2
x_4 x_2 x_2 x_2
x_6 x_6
x_6 x_7
x_2 x_3
x_1
IRSS test projects
y_9: temperature-rise test
x_5 x_5 x_5 x_5 x_5 x_5
y_{11}: efficiency test
x_5 x_5 x_5 x_5 x_5 x_5
y_7: load test
x_5 x_5 x_5
y_1: ripple torque coefficient
x_5
y_6: no-load test
x_2 x_2 x_2 x_2 x_2 x_2
y_2: speed fluctuation coefficient
x_2 x_2
y_{10}: overspeed test
x_4 x_2 x_2 x_2
y_4: electrical time constant
x_6 x_6
y_8: locked-rotor test
x_6 x_7
y_3: positive and negative slip
x_2 x_3
y_5: safety test
x_1

Figure 9: Schematic diagram of the parallel testing result of application case.

Table 3: Optimizing schedule of the optimization model of application case based on the SA algorithm.

Times of cooling	Temperature T	Times of state transitions	Test time (min)	Ratio of optimization time (%)
0	100	—	273	—
5	77.38	9	173	63.36
10	59.87	6	148	54.21
15	46.33	3	148	54.21
20	35.84	4	138	50.55

Table 4: Initialization parameters of the solving algorithm for comparison experiment.

Algorithm	Initialization parameters				Maximum iteration times	Test no.
	P_c	P_m	popSize	Gen_iter		
GA	0.9	0.5	20	50	1000	1
			50	50	2500	2
			100	100	10000	3
SA	T_k	T_f	lam	res	900	4
	100	10	0.95	20		

Table 5: Comparative analysis results of the comparison experiment.

Algorithm	Optimal solution (min)	Test no.	Times of occurrence of optimal solution	Average (min)	SD	Average running time (s)
GA	138	1	10	146.6	4.8518	0.073232
		2	27	142.9	3.6729	0.174638
		3	54	140.35	2.5937	0.544116
SA		4	73	139.5	2.5981	0.031773

population size popSize and the number of genetic iterations Gen_iter are different. The reason for the setting of these combinations of initialization parameters is that the population size and the number of genetic iterations of the GA determine the maximum iteration times of one time algorithm running, where the maximum iteration times of the GA are popSize $*$ Gen_iter, while the maximum iteration times of the SA algorithm are $\left\lceil \log_{lam}^{(T_f/T_k)} \right\rceil \times$ res. In this way, it is easy to compare the performance of these two algorithms based on the maximum iteration times. We use the GA and the SA algorithms to solve the optimization model of application case based on the initialization parameters shown in Table 4. As seen from Table 4, four tests are implemented, and the maximum iteration times of test no.1 are similar to those of test no.4. In each test, we run 100 times of the algorithm, and the solutions obtained in the 100 times of algorithm running of each test are compared and analyzed based on the times of occurrence of optimal solution, average value, standard deviation value, and the average running time. Table 5 shows the comparative analysis results of the comparison experiment. As can be concluded from Table 5, the overall performance of the SA algorithm is better than that of the GA. When the maximum iteration times are similar, the solving performance of the SA algorithm is obviously superior to the GA. Although the performance of the GA gets better with the increase of the population number and the number of genetic iterations, it needs 10 times of maximum iteration times for the GA to achieve the performance similar to the SA algorithm. However, the larger the maximum iteration times is, the longer running time the algorithm needs. And the large population size would bring high space complexity of the GA algorithm. Therefore, the SA algorithm is an ideal solving method to the optimization model of IRSS test procedure on the IITP.

7. Conclusions

In this paper, we proposed a test procedure optimization method for the industrial robot servo system on the integrated testing platform. In the proposed method, firstly, we creatively defined the execution subelements of the integrated platform as well as the industrial robot servo system test project ordered sequences. In addition, we designed an "$\cap$" operation to dig out the optimization relationships between the industrial robot servo system test project sequences. Finally, based on these optimization relationships, we established an optimization model for the industrial robot servo system test procedure on the integrated platform, and by solving the model, we obtained the optimal test order of the industrial robot servo system test projects with shortest test time when testing on the integrated testing platform. Therefore, the test efficiency of the integrated testing platform improved. The case analysis showed that the proposed method can shorten the test time of the industrial robot servo system by 50.55%. The comparative experiment also showed the superiority of the proposed method.

Moreover, the proposed method can be applied to similar scenarios that test samples with multiple test projects are tested on the integrated testing platform. Besides, the ideas to apply our proposed method to deal with the multiple stress combination experiments in battery aging test [36] and the establishment of new models for complex systems based on the process similarity [37] are considered in our future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Guangdong High-End Equipment Manufacturing Plan Project under grant no. 2017B090914003.

References

[1] T. Brogårdh, "Robot control overview: an industrial perspective," *Modeling, Identification and Control: A Norwegian Research Bulletin*, vol. 30, no. 3, pp. 167–180, 2009.

[2] T. Yuan, D. Wang, X. Wang, X. Wang, and Z. Sun, "High-precision servo control of industrial robot driven by PMSM-DTC utilizing composite active vectors," *IEEE Access*, vol. 7, pp. 7577–7587, 2019.

[3] D. B. Zhu, J. W. Han, X. J. Wang, Y. P. Zhu, and X. B. Guo, "Model and performance study on the PMLSM," in *Proceedings of the IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 3670–3675, New York, NY, USA, 2017.

[4] H. Guo, Y. F. Li, and X. Zhang, "Development of a performance test platform of direct-drive servo valve linear force motor," in *Proceedings of the 2018 15th International Conference on Ubiquitous Robots*, pp. 781–785, New York, NY, USA, 2018.

[5] J. Q. Yang, D. Y. Wang, and W. H. Zhou, "Precision laser tracking servo control system for moving target position measurement," *Optik*, vol. 131, pp. 994–1002, 2017.

[6] H. Wang, Z. Y. Zhang, J. Long, L. Wang, and X. R. Ye, "Reliability evaluation method for robot servo system based on accelerated degradation test," in *Proceedings of the 2019 22nd International Conference on Electrical Machines and Systems*, pp. 1942–1947, New York, NY, USA, 2019.

[7] S. J. George, K. S. S. Ranjini, C. Rajagopalan, and S. Murugan, "Design and development of automated high temperature motor test facility," *Sadhana*, vol. 45, no. 1, p. 6, 2019.

[8] L. Sidhom, M. Smaoui, and X. Brun, "Tracking position control of electrohydraulic system with minimum number of sensors," *Studies in Informatics and Control*, vol. 25, no. 4, pp. 479–488, 2016.

[9] X. Zhang, H. He, J. Zheng, and H. Dong, "Servo mechanism test system based on PXI," *Ordnance Industry Automation*, vol. 35, no. 12, pp. 1–3, 2016.

[10] M. D. Luo, Z. Q. Peng, and J. Y. Li, "Design of a four-channel automatic test system for servos," in *Proceedings of the 2016 IEEE Chinese Guidance, Navigation and Control Conference*, pp. 791–795, New York, NY, USA, 2016.

[11] Z. Cheng, A. Chen, and J. Li, "Analysis of simulate model based on servo mechanism simulation device," in *Proceedings of the 2019 IEEE 8th International Conference on Fluid Power and Mechatronics*, pp. 122–126, Wuhan, China, April 2019.

[12] M. W. Ingalls and D. R. Broussard, "Modular design of test systems," in *Proceedings of the 17th Capacitor and Resistor Technology Symposium CARTS'97*, pp. 251–254, Jupiter, FL, USA, March 1997.

[13] N. Christofides, "The travelling salesman problem," *Combinatorial Optimization*, pp. 131–149, 1979.

[14] B. Yang, W. Li, J. Wang, J. Yang, T. Wang, and X. Liu, "A novel path planning algorithm for warehouse robots based on a two-dimensional grid model," *IEEE Access*, vol. 8, pp. 80347–80357, 2020.

[15] B. Wei, Y. Wei, Y. Wang, and X.-H. Cao, "Image mosaic based on TSP," in *Proceedings of the International Conference on*

Mechanism Science and Control Engineering, pp. 364–367, Changsha, China, 2014.

[16] D. Schermer, M. Moeini, and O. Wendt, “A branch-and-cut approach and alternative formulations for the traveling salesman problem with drone,” *Networks*, vol. 76, no. 2, pp. 164–186, 2020.

[17] G. Campuzano, C. Obreque, and M. M. Aguayo, “Accelerating the Miller-Tucker-Zemlin model for the asymmetric traveling salesman problem,” *Expert Systems with Applications*, vol. 148, 2020.

[18] T. Moyaux and E. Marcon, “Cost of selfishness in the allocation of cities in the multiple travelling salesmen problem,” *Engineering Applications of Artificial Intelligence*, vol. 89, 2020.

[19] X. Meng, J. Li, X. Dai, and J. Dou, “Variable neighborhood search for a colored traveling salesman problem,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1018–1026, 2018.

[20] M. Shahmanzari, D. Aksen, and S. Salhi, “Formulation and a two-phase matheuristic for the roaming salesman problem: application to election logistics,” *European Journal of Operational Research*, vol. 280, no. 2, pp. 656–670, 2020.

[21] G. Aifadopoulou, G. Tsaples, J. M. S. Grau, I. Mallidis, and N. Sariannidis, “Management of resource allocation on vehicle-sharing schemes: the case of Thessaloniki’s bike-sharing system,” *Operational Research*, 2020.

[22] D. Delle Donne, V. Di Tomaso, and G. Duran, “Optimizing leaf sweeping and collection in the Argentine city of Trenque Lauquen,” *Waste Management & Research*, 2020.

[23] W. J. Cook, *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*, Princeton University Press, Princeton, NJ, USA, 2011.

[24] A. Ahamdi-Javid and N. Ramshe, “Designing flexible loop-based material handling AGV paths with cell-adjacency priorities: an efficient cutting-plane algorithm,” *Quarterly Journal of Operations Research*, vol. 17, no. 4, pp. 373–400, 2019.

[25] R. Salman, F. Ekstedt, and P. Damaschke, “Branch-and-bound for the precedence constrained generalized traveling salesman problem,” *Operations Research Letters*, vol. 48, no. 2, pp. 163–166, 2020.

[26] Y. Yuan, D. Cattaruzza, M. Ogier, and F. Semet, “A branch-and-cut algorithm for the generalized traveling salesman problem with time windows,” *European Journal of Operational Research*, vol. 286, no. 3, pp. 849–866, 2020.

[27] I. Kucukoglu, R. Dewil, and D. Cattrysse, “Hybrid simulated annealing and tabu search method for the electric travelling salesman problem with time windows and mixed charging rates,” *Expert Systems with Applications*, vol. 134, pp. 279–303, 2019.

[28] Q. Liu, S. Du, B. J. van Wyk, and Y. Sun, “Niching particle swarm optimization based on Euclidean distance and hierarchical clustering for multimodal optimization,” *Nonlinear Dynamics*, vol. 99, no. 3, pp. 2459–2477, 2020.

[29] S. Mitra and D. Das, “An efficient VLSI test data compression scheme for circular scan architecture based on modified ant colony meta-heuristic,” *Journal of Electronic Testing*, vol. 36, no. 3, pp. 327–342, 2020.

[30] E. Bas and E. Ulker, “Discrete social spider algorithm for the traveling salesman problem,” *Artificial Intelligence Review*, 2020.

[31] A. G. Hussien, A. E. Hassanien, E. H. Houssein, M. Amin, and A. T. Azar, “New binary whale optimization algorithm for discrete optimization problems,” *Engineering Optimization*, vol. 52, no. 6, pp. 945–959, 2020.

[32] Y. Hao, F. Qiang, Y. Jiaqi, and Z. Caiming, “An improved chicken swarm optimization for TSP,” in *Proceedings of the International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019*, pp. 211–220, Huainan, China, 2020.

[33] V. Fabian, “Simulated annealing simulated,” *Computers & Mathematics with Applications*, vol. 33, no. 1-2, pp. 81–94, 1997.

[34] K.-W. Yeh, J.-L. Huang, and L.-T. Wang, “CPP-ATPG: a circular pipeline processing based deterministic parallel test pattern generator,” *Journal of Electronic Testing*, vol. 32, no. 5, pp. 625–638, 2016.

[35] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Boston, MA, USA, 1989.

[36] X. Tang, K. Liu, X. Wang, F. Gao, J. Macro, and W. D. Widanage, “Model migration neural network for predicting battery aging trajectories,” *IEEE Transactions on Transportation Electrification*, vol. 6, no. 2, pp. 363–374, 2020.

[37] J. Lu, K. Yao, and F. Gao, “Process similarity and developing new process models through migration,” *AIChE Journal*, vol. 55, no. 9, pp. 2318–2328, 2009.