International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 4916-4925

https://doi.org/10.70517/ijhsa464335

Cooperative Optimization of Industrial Robots and Electrical Drive Systems Based on Multi-level Genetic Algorithms

Xiaoying Yan[1,*]
[1] College of Engineering, Caofeidian College of Technology, Tangshan, Hebei, 063200, China
Corresponding authors: (e-mail: Yanxiaoying L L@163.com).

Abstract Aiming at the energy efficiency optimization difficulties caused by the multi-component coupling and hierarchical structure of industrial robot electrical drive systems, this paper proposes a multilevel genetic algorithm (MGA) co-optimization method. First, an improved dq-axis motor model integrating iron loss, saturation effect and temperature influence is established to define a multi-constraint optimization problem with the objective of minimizing the total energy consumption of the system (covering the motor, inverter and transmission loss). Second, a hierarchy-dependent genetic coding scheme is designed to express the variable structure design space through hierarchical description with prefix tagging method, and the adapted genetic operators are developed. In ZDT1/3/4 tests, the MGA improves the hypervolume (HV) by 3.3%~6.2% compared with the conventional GA, increases the independent solution ratio by 4.5%~7.1%, and reduces the generation distance (GD) and inverse generation distance (IGD) by up to 72% (e.g., the IGD of ZDT4 is reduced from 0.0304 to 0.0084). In the drive system layout optimization, the convergence speed of MGA is improved by a factor of 2.7 over GA with objective function values of 4.394 and 4.311, respectively. Based on the multi-electrical aircraft load management experiments, the system achieves 98.66% energy efficiency under healthy working conditions, 35kW load shedding by priority optimization when the main generator fails, and a 21-fold improvement in the computational efficiency of the hierarchical control strategy (23.41 seconds vs. 8.35 minutes for a single layer).

Index Terms industrial robot, electrical drive, multilevel genetic algorithm, multi-constraint optimization

I. Introduction

Electrical drive systems play an increasingly important role in industry, transportation and daily life. In the production process of industrial robots, the electrical drive system, as a core component, has a direct impact on the overall performance and market competitiveness of industrial robots [1], [2]. In addition, improving the energy efficiency of electrical drive systems not only reduces operating costs, but also reduces carbon emissions, which is important for realizing sustainable development [3], [4]. Traditional optimization methods are often difficult to cope with the complexity and nonlinear characteristics of electrical drive systems, while genetic algorithms, as a kind of heuristic optimization method with powerful global search capability and adaptability to complex problems, provide new possibilities to solve this challenge [5]-[8].

Genetic algorithm can effectively avoid local optimum by simulating the biological evolution process and using population search and stochastic operation [9]. It does not depend on the specific mathematical model of the problem and can deal with all kinds of optimization problems, and this flexibility is especially suitable for complex engineering problems [10], [11]. In addition, the genetic algorithm has good parallelism and computational efficiency, is insensitive to the initial conditions and parameter settings, and has strong fault tolerance, and this robustness is particularly important in practical engineering applications [12]-[14]. Therefore, through the design and validation of innovative algorithms, it is expected to develop an optimization technique that can significantly improve the energy efficiency of the system, and provide a theoretical basis and practical guidance for the intelligence and high efficiency of electrical drive systems.

This paper proposes and constructs a cooperative optimization methodology system based on multilevel genetic algorithm (MGA), aiming to achieve the global energy efficiency optimization of industrial robot electrical drive system under the severe engineering constraints. An underlying physical model that accurately reflects the energy flow and constraints of the system is firstly established as the optimization basis. The model innovatively integrates key factors such as iron loss (through parallel iron loss resistors), saturation effect (dq-axis inductance expressed as a nonlinear function of current), and temperature effect (correction of permanent magnet remanence and stator resistance), which significantly improves the prediction accuracy of the model under complex operating conditions. Second, for the inherent hierarchical structural characteristics of the system (e.g., component selection, substructure configuration), a matching multilevel genetic coding scheme is designed to effectively express and manipulate the structurally variable design space. The technique employs a hierarchical description and prefix labeling method to accurately describe the tree-like hierarchical structure of the system. Individuals are then represented as selected gene sequences, where each gene represents a design variable for a selected substructure. Finally, genetic operators adapted to this coding scheme are designed to ensure that the optimization process efficiently searches for feasible solutions with different structural complexities. The selection operator employs league selection to retain superior individuals by randomly comparing their fitness. The crossover operator is designed with full consideration of the hierarchical dependence of genes, when two parent individuals exchange a certain substructure, all the genes of the offspring under them must be exchanged as a whole as well, which ensures the structural integrity of the offspring. The operation of the mutation operator starts from the highest level of the hierarchical structure and proceeds down the hierarchy. Mutation may change the parent gene, which causes all child genes under it to be reset or reselected to fit the new parent structure, thus effectively exploring the design space for different structural complexities. Together, these operators guarantee the optimization-seeking ability of MGA in complex hierarchical problems.

II. Construction of cooperative optimization method for electrical drive system based on multilevel genetic algorithm

II. A.2.1 Motor Modeling and Optimization Objective Definition

In order to accurately assess the energy efficiency of an electrical drive system, an accurate motor model was first developed in this study. A modified $d-q$ -axis model is adopted, which not only considers the basic electromagnetic relationship, but also incorporates factors such as iron loss, saturation effect and temperature influence. Specifically, iron loss is introduced by connecting an iron loss resistor in parallel in the $d-q$ -axis equivalent circuit; the saturation effect is described by expressing the $d-q$ -axis inductance as a nonlinear function of the current; and the temperature effect is mainly manifested in the variations of permanent magnet spurs and stator resistance, and temperature coefficients are employed to correct these parameters. This comprehensive modeling approach enables a more accurate prediction of motor performance and losses under various operating conditions.

Based on this, the optimization objective and constraints are defined. The optimization objective is to minimize the total system energy consumption, including motor losses (copper, iron, and mechanical losses), controller losses (switching losses, conduction losses), and drive train mechanical losses. This multi-objective optimization problem can be expressed as

$$\min J = \int (p_{cu} + p_{fe} + p_{\text{mod}\,a} + p_{inv} + p_{trane})dt \tag{1}$$

Here, P_a is the copper loss; P_{fo} is the iron loss; P_{avch} is the mechanical loss of the motor; P_{inv} is the inverter loss: P_{uran} is the transmission system loss. The constraints include (1) the motor temperature rise does not exceed the allowable value $T \le T_{\max}$; (2) the output torque meets the load requirements $T_e \ge T_{loal}$; (3) the system stability meets the requirements: the real part of all the poles should be negative: and (4) the currents and voltages do not exceed the rated values $I \le I_{ranl}, V \le V_{rand}$. This comprehensive definition of the objective function and constraints ensures that the optimization results not only improve energy efficiency, but also satisfy various requirements in practical engineering.

II. B.2.2 Characterization of multilevel gene coding

Based on the above precise motor model and multi-objective and multi-constraint optimization problem definition, an optimization algorithm that can effectively deal with its inherent hierarchical structure and variable dependencies is urgently needed to achieve the global optimization of the electrical drive system of industrial robots. However, traditional optimization coding methods are difficult to adapt to such complex design spaces, which leads to the need for multilevel genetic coding techniques.

Traditional coding methods are generally expressed in terms of one- or multi-dimensional matrices, which are not applicable in design problems with multi-level structures. Design problems with multilevel structures usually have many different structures layered at the same sublayer with multiple options, and some structures have some lower level structures ...... These different levels of structure and the existence of inheritance and crossover relationships, different levels of the design variables change will certainly make its related design variables with the change. In this way, the crossover and mutation operations of traditional methods become infeasible in multilevel design problems. Therefore, we propose a genetic coding technique with multiple levels.

In the following, we explore multilevel genetic coding through a simple mechanical product design problem with a multilevel structure:

The schematic of a mechanical product with multilevel structure is shown in Fig. 1, this product consists of three building blocks A, B, and C. Among them, building block A has three optional designs, A1-1, A1-2, and A1-3; building block B has two optional designs, B1-1 and B1-2: building block C has four optional designs, C1-1, C1-2, C1-3, and C1-4. Among them, B1-2 has lower-price level sub-structures x, y; sub-structure x in turn has two optional designs, x1-1, x1-2, and sub-structure y has two optional designs, y1-1, y1-2; among them, y1-2 has lower-level optional designs, a1, a2, a3, a4.

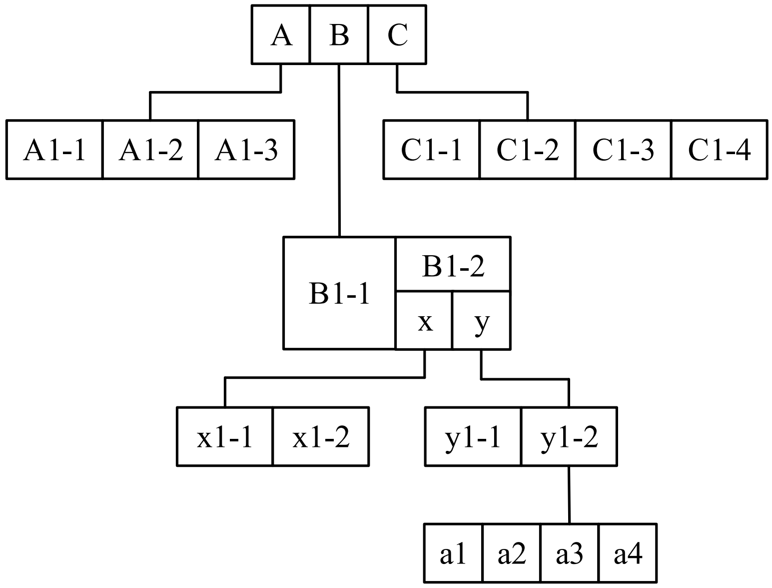

Figure 1: Schematic diagram of mechanical products with multi-level structure

Based on this mechanical system, we provide a detailed description of the multilevel genetic code. In the first step, the first layer is identified first, this system consists of three substructures A, B and C. In the third step, we are going to divide the first level of structure at a lower level, where structure A has three optional designs, structure B has two optional designs and C has four optional designs. We describe them (3, 2, 4) and define that all the optional designs of the substructures are denoted by (), and these optional designs constitute our second level of structure. B1-2 has two substructures of lower order x, y. In the third step, identify the substructure of lower order in the second level of the structure, where x has 2 optional designs and 2 has two optional designs. At this point we describe it as (2, 2), and since it belongs to the second optional design in the substructure B1-2 of layer 1, we add the prefix 1, 2- in front of (2, 2), which is used to indicate that it is the second optional design in the substructure B1-2 of layer 1, and this is the third layer of the structure that we want to look for: step 4, analyze the third layer of the structure, and find the lower-order structure of y1-2 having the options, and these four options make up our fourth level of structure. At this point we describe it as 1, 2-2, 2-(4).

When facing more general mechanical systems, if a sub-structure has N lower substructures, each of which in turn has X_{j-i} (j is the number of layers in which it is located, $i=1,2,...,N$) for an optional design, we denote it as $(X_{j-1}, X_{j-2},...)$. If this substructure is the t th optional design belonging to the s th level of substructures, we prefix $(X_{j-1}, X_{j-2},...)$ preceded by the prefix $s,t-$. When there are higher level substructures, we add more prefixes in front of them to indicate the positional relationship of the current substructure.

Once all the structures in the system are described, we can use point nodes to describe the location of all the structures. If a node A is located on top of another node B, call this node A the parent of node B, and call B the child of node A. This way we can get the multi-level information of this mechanical product clearly and connect the nodes with lines. In this way, we can get the multi-level information of this mechanical product and connect the nodes with lines. This method is called the structural description method of the multilevel structure system.

After defining the structural description method of the system, we need to define the individual description method. If an individual of our previous example is shown in Fig.

In this example, A1-3, B1-2 and C1-1 are selected as the design substructure of the first level, denoted as [3, 2, 1], and the definition [] denotes the selected design substructure among all the optional designs of the substructure. This selected substructure is a design variable in the ontology and is defined as a gene. A complete system has an infinite number of selected substructures, i.e., the structure of a multilevel system is represented by a set of genes. In the substructure of B1-2 we select x1-1, y1-2 as its selected design, denoted as [1, 2], which belongs to the 2nd design substructure of the 1st level, at this point we add the prefix 1, 2- before |1, 2|. Similarly, the last gene in this individual is denoted as 1, 2-2, 2-[2].

We generalize this approach to the general design problem. If a substructure has N lower level substructures and a bi design is chosen from its optional designs, this is denoted as a gene [b1, b2, ...]. If this gene belongs to the t-th selected design of the s-th level substructure when the gene [b1, b2, ...] preceded by the prefix s, t-. More prefixes must be added to the genes when there is a higher level of substructure for this substructure. In this way a complex individual with a multilevel structure is represented. If a gene A is located one level above another gene B, the footnote A is the parent gene of B, and B is the child gene of A.

II. C.2.3 Genetic operators for multilevel genetic algorithms

Having defined a genetic coding scheme capable of accurately describing the multilevel design variables and their dependencies for industrial robot drive systems, the design of matching genetic manipulation operators is required to ensure that genetic algorithms can efficiently search and optimize in non-fixed-length, hierarchically-associated solution spaces defined by structural variability. Therefore, the design of selection, crossover, and mutation operators adapted to multilevel genetic coding is detailed below.

II. C. 1) 2.3.1 Selection of operators

The league selection algorithm is used in the system, and its basic idea is: randomly select a certain number of individuals (usually 2) from the group, select the one with the highest fitness as the next generation of individuals, and repeat the execution until the number of selected individuals reaches a predetermined value.

II. C. 2) 2.3.2 Intersection operators

Since the design state space is represented by multiple levels of gene coding, access to lower level gene design variables is dependent on higher level gene design variables. When the higher-level gene design variables change, the lower-level design variables must also change, and individual gene lengths may also change. The crossover operation, i.e., interchanging the corresponding genes between two individuals, is a process whereby when the selected genes of one substructure are exchanged with another, all the corresponding exchange operations in the next generation follow accordingly. The process of crossover operation is shown in Figure 2.

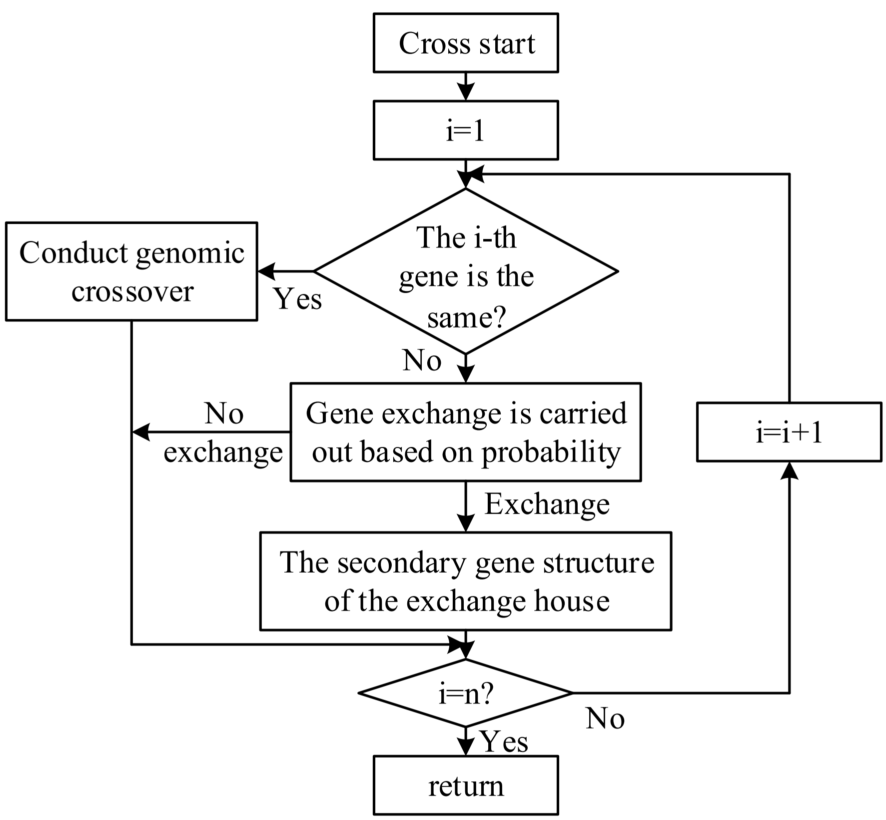

Figure 2: Flowchart of cross-operation

II. C. 3) 2.3.3 Variational operators

In mutation operation, mutation starts from the highest level of the multilevel structure. The mutation operation is invoked in the same way as the crossover operation, acting in the subgenome, and the process of the mutation operation is shown in Figure 3.

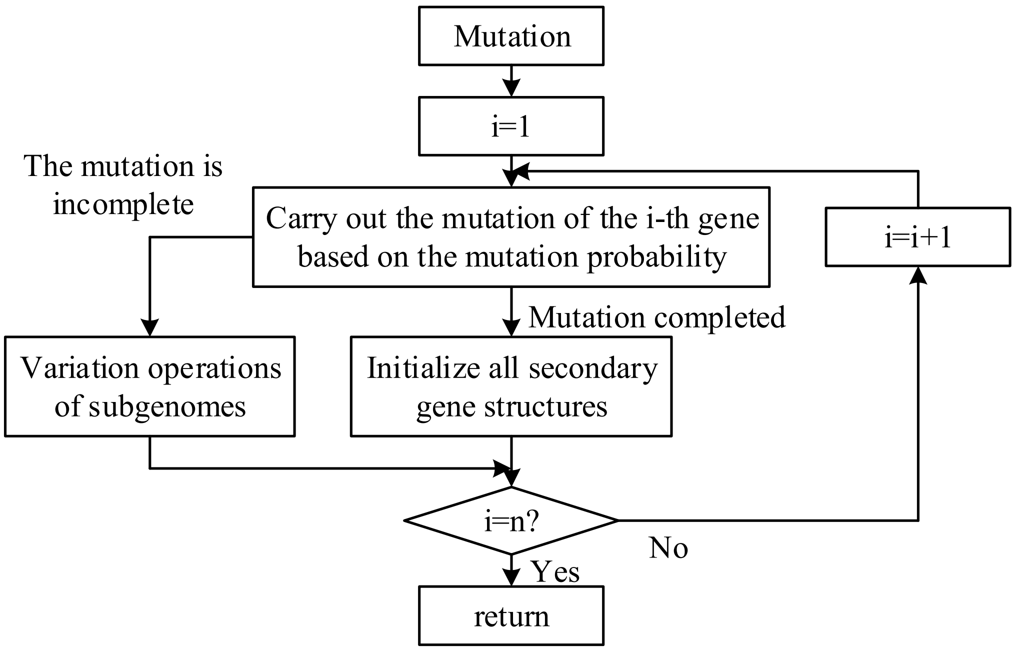

Figure 3: Flowchart of variation operation

III. Hierarchical genetic algorithm performance validation and industrial robot drive system optimization experiments

Based on the above constructed multi-level genetic algorithm method system, in order to verify its practical effectiveness in industrial robot electrical drive system optimization, this chapter will carry out multi-dimensional experimental validation: firstly, quantitatively assess the performance advantage of the algorithm through the standard test function, and then apply it to the drive system layout optimization, and finally, combine with the simulation of the typical working conditions to verify the practicality of the project.

III. A. 3.1 Based on improved multilevel genetic algorithm test

In this chapter, a multilevel genetic algorithm is used to solve the electrical drive system optimization problem, and the superiority of the proposed multilevel algorithm is demonstrated through test functions.

III. A. 1) 3.1.1 Evaluation of performance indicators

In order to test the performance of the multilevel genetic algorithm, the test basis functions ZDT1, ZDT3 and ZDT4 are used to test the performance of the algorithm in this chapter.

The optimization objectives of ZDT1, ZDT3 and ZDT4 are shown in equations (2)-(4):

$$ZDT1=\begin{cases}\min F_1(x)=x_1\\ \min F_2(x)=G\left(1-\sqrt{\dfrac{F_1}{G}}\right)\\ G(x)=1+9\sum_{i=2}^{m}\dfrac{x_i}{m-1}\\ s.t.0\le x_i\le 1, i=1,2,\dots,30\end{cases} \tag{2}$$

$$ZDT3=\begin{cases}\min F_1(x)=x_1\\ \min F_2(x)=G\left[1-\sqrt{\dfrac{F_1}{G}}-\dfrac{F_1}{G}\sin(10\pi F_1)\right]\\ G(x)=1+9\sum_{i=2}^{m}\dfrac{x_i}{m-1}\\ s.t.0\le x_i\le 1,\ i=1,2,\ldots,30\end{cases} \tag{3}$$

$$ZDT4=\begin{cases}\min F_1(x)=x_1\\ \min F_2(x)=G\left(1-\sqrt{\dfrac{F_1}{G}}\right)\\ G(x)=1+10(m-1)+\sum_{1=2}^{m}\left[x_1^2-10\cos\left(4\pi x_1\right)\right]\\ s.t.0\le x_1\le 1,-10\le x_1\le 10, i=2,\ldots,10\end{cases} \tag{4}$$

For the optimization results of the multilevel genetic algorithm, the following four metrics are selected in this chapter to evaluate the performance of the algorithm.

The first metric is the hypervolume metric (HV).HV is the space surrounded by the nondominated solution and the reference point obtained by the algorithm. It is expressed as the area enclosed by the reference point and the nondominated solution in the two-dimensional objective, and as the Euclidean volume of the space enclosed by the reference point and the nondominated solution in higher dimensions. The larger this metric is, the closer the entire set of nondominated solutions is to the coordinate axis, i.e., closer to the Pareto front, which represents better algorithmic performance.The HV is computed as shown in Equation (5):

$$HV=\delta\left(\bigcup_{i=1}^{|S|}v_i\right) \tag{5}$$

where δ - Leberger measure, S - the set of non-dominated solutions, v_i - the hypervolume formed by the non-dominated solution set and the reference point

The second metric is the proportion of independent solutions, the number of independent solutions in the final solution set as a proportion of the entire population size for the same population size. This metric evaluates the repeatability of the solution set, so the larger this ratio is, the greater the diversity of the resulting solution set, i.e., the better the performance. The proportion of independent solutions is calculated as shown in equation (6):

$$R_{independent}=\frac{N_{independent}}{N_{population}}\times 100\% \tag{6}$$

where $N_{independent}$ - number of independent solutions, $N_{population}$ - population size

The third metric is Generation Distance (GD), which is the average minimum distance from each point in the solution set to a point in the true Pareto boundary, as shown in equation (7):

$$GD(P,P^*)=\frac{\sqrt{\sum_{y\in P}\min_{x\in P^*}dis(x,y)^2}}{|P|} \tag{7}$$

The fourth metric is the Inverse Generation Distance (IGD), which is the average distance between each point in the true Pareto boundary to its nearest solution, as shown in equation (8):

$$IGD(P,P^*)=\frac{\sum_{x\in P^*}\min_{y\in P}dis(x,y)}{\left|P^*\right|} \tag{8}$$

The pair of metrics GD and IGD both describe the distance of the solution set from the true Pareto frontier, the difference being that one starts from the solution set and the other finds the distance to the other side from a uniform set of points on the true Pareto frontier. Both metrics are such that the shorter the distance, the closer the solution set is to the true Pareto solution set, i.e., the better the algorithm performs.

III. A. 2) 3.1.2 Comparison of performance test results

For the traditional genetic algorithm GA and multilevel genetic algorithm MGA tested performance metrics are compared as shown in Table 1.

Table 1: Comparison of algorithm performance tests

Test function	HV		Rindependent		GD		IGD	
	GA	MGA	GA	MGA	GA	MGA	GA	MGA
ZDT1	0.5713	0.5902	53.48%	58.95%	0.0253	0.0119	0.0249	0.0053
ZDT3	4.2641	4.5261	56.21%	60.68%	0.0374	0.0236	0.0765	0.0434
ZDT4	3.5253	3.7075	54.56%	57.93%	0.0257	0.0141	0.0304	0.0084

From the comparison in Table 1, it can be clearly found that the multilevel genetic algorithm MGA shows better performance than the traditional genetic algorithm GA in all the ZDT series of test functions. In the hypervolume metric HV, MGA is 0.5902, 4.5261 and 3.7075 in ZDT1, which are about 3.3%, 6.2% and 5.2% higher than GA's 0.5713, 4.2641 and 3.5253, respectively, indicating that its solution set is closer to the Pareto frontier and has a wider coverage. Among the independent solution ratios, the solution set diversity of MGA is significantly higher (ZDT1: 58.95% vs. 53.48%; ZDT3: 60.68% vs. 56.21%; ZDT4: 57.93% vs. 54.56%).The generation distance GD and the inverse generation distance IGD are lower than that of GA across the board for MGA.For example, in ZDT4, the GD decreases from 0.0257 to 0.0141, a decrease of 45%, and IGD decreases from 0.0304 to 0.0084, a decrease of 72%, which verifies that the MGA solution set has a higher degree of closeness to the true Pareto frontier.

All in all, MGA outperforms GA in terms of multi-objective optimization ability (coverage, diversity, convergence), and the advantage is especially more significant when dealing with complex nonlinear problems.

Meanwhile, Fig. 4 demonstrates the comparison of the solution sets of the two algorithms for the three test basis functions.

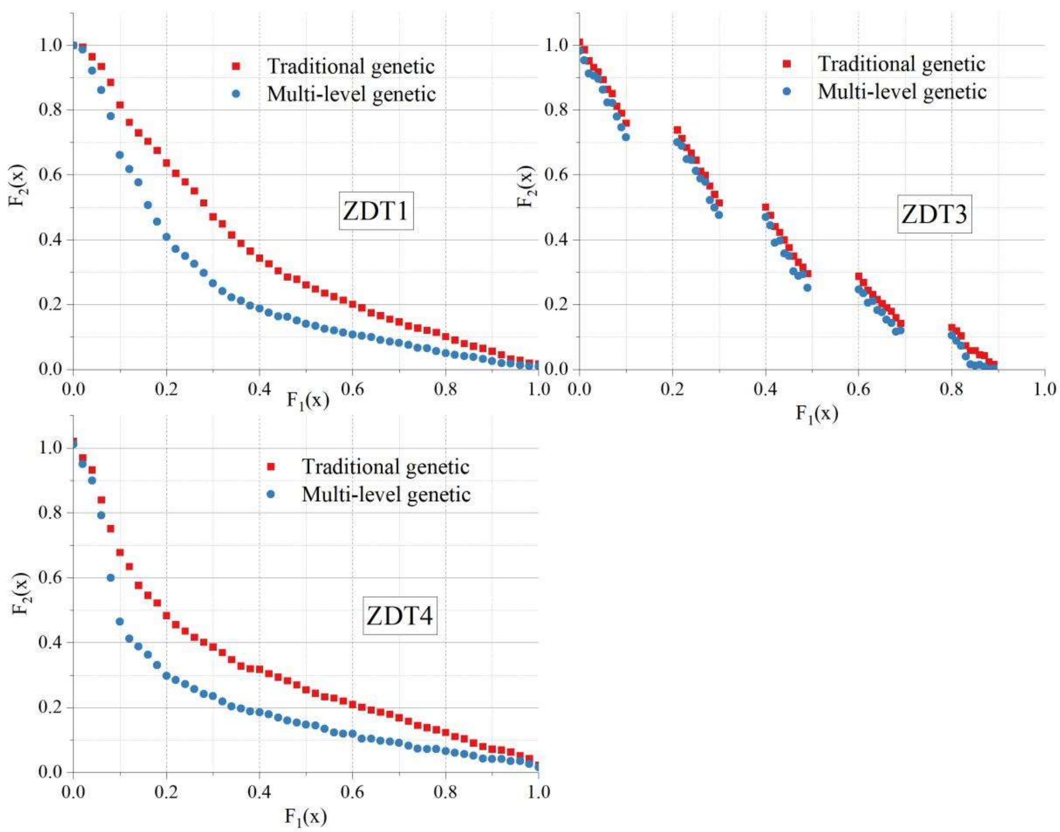

Figure 4: The solution sets of two algorithms for three test basis functions

Whether it is ZDT1, ZDT3 or ZDT4 under the three middle test basis functions and four metrics, the solution set obtained by the multilevel genetic algorithm is closer to the coordinate axes than the traditional genetic algorithm, i.e., it has better performance.

III. B. 3.2 Electrical drive system layout optimization test

After verifying the theoretical advantages of the algorithm in standard test functions, its applicability to practical engineering problems needs to be further examined. In this section, the MGA is applied to the layout optimization of electrical drive systems to compare its difference with the traditional GA in terms of iterative convergence and solution quality.

After completing the coding of the operating elements, the construction of the mathematical model, and the calculation of the relevant data. Input the data into the genetic algorithm and set the relevant control parameters, and use MTLAB software to solve the problem, and the running environment is c language. For the design of this interface layout, the relevant parameters of the genetic algorithm are set as follows: population size NIND=100, crossover rate PC=0.8, mutation rate Pm=0.090, generation gap GGAP=0.90, and the maximum number of iterations GENMAX=200. The two algorithms were run for 20 times respectively, and the optimization results of the traditional genetic algorithm and the multi-level genetic algorithm were compared, and the iterative convergence curves are shown in Figure 5.

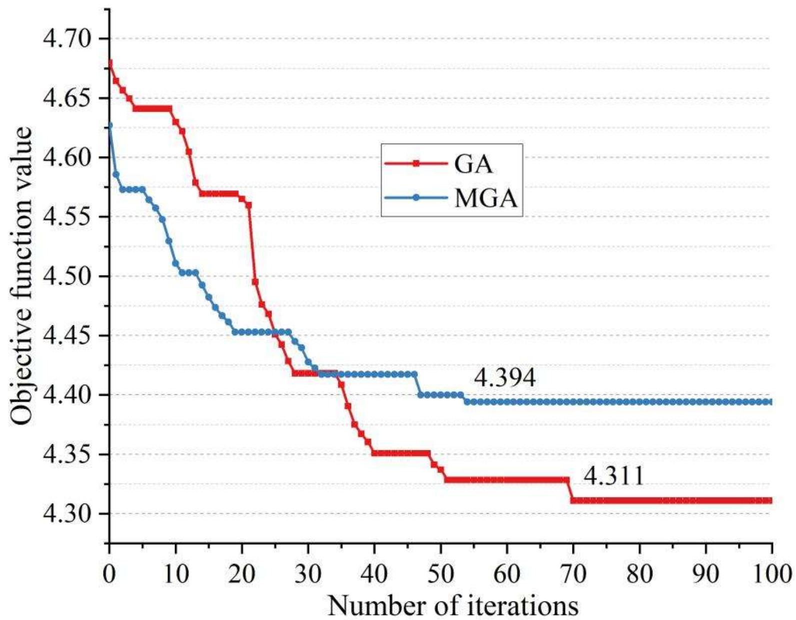

Figure 5: Comparison of the iterative curves of GA and MGA

From the running results, we can find that in the same layout optimization problem, the objective function value of GA is 4.311, while the objective function value of MGA is 4.394, and the optimization result of GA is better than MGA, while the solution speed of MGA is better than that of GA, and we also therefore validate that we conclude that at the initial stage, GA is slow in finding the optimum, but has a strong global searching ability, and MGA is able to search for the optimum quickly at the initial stage, and therefore also easy to cause the optimization to fall into local optimum.

III. C. 3.3 Simulated Load Experiments under Different Operating Conditions

In addition to static layout optimization, the dynamic load management capability of the electrical drive system is crucial. For this reason, in this section, the effectiveness of the proposed method in dynamic load scheduling and energy efficiency control is verified through simulation experiments by combining the typical working conditions of the V-tail system of a multi-electric airplane.

The main working conditions studied in the simulation experiments are (1) a normal power system with healthy loads and (2) a healthy load but the main generator fails. Next, each case is described in detail and the results are analyzed.

III. C. 1) 3.3.1 Normal power system with healthy loads

Example 1: In task phase 1, all loads need to be powered according to the task requirements. The total power requirement of the electrical system is 72.84 kW, all of which is provided by the main generator. In task phase 2, loads 2 and 6 are stopped as required by the task. Before 0.3 s, the values of S1-S9 are all 1, indicating that the switching state of all loads is closed. After 0.3 s, the values of S2 and S6 are all 0, indicating that the switches of loads 2 and 6 are disconnected.

The power supply and total power consumption of all loads are shown in Fig. 6. The red line in the figure indicates the generator output power and the blue line indicates the total power of all loads. Before 0.3s, the average power provided by the generator is PG = 72.53 kW, of which the sum of all the loads consumes Pi = 71.56 kW, and the rest of the power is mainly dissipated in the transmission process. However, the losses are small and the efficiency reaches 98.66%. In task phase 2, the total generator output and total load consumption are PG=56.48kW and Pi=55.64kW respectively, and the efficiency reaches 98.51%. A small oscillation occurs at 0.3s off loads 2 and 6, which is determined by the combination of the rectifier gain and time constant settings in the GCU control law.

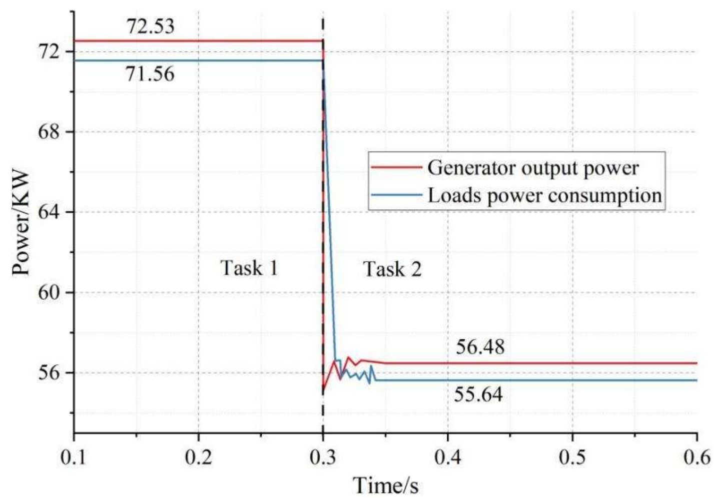

Figure 6: The output power and power consumption of the load in Example 1

III. C. 2) 3.3.2 Failure of the main generator with a healthy load

Calculation example 2: Assume that the main generator fails at 0.15s and the primary AC bus is taken over by a 40kW APU. The task requirements of the loads are kept the same as in arithmetic example 1, i.e., task phase 1 before 0.3s and considered task phase 2 after that.

The electrical power consumed by all loads and the total power are shown in Fig. 7. The red line in the figure indicates the generator output power and the blue line indicates the total power of all loads. During the normal operation of the main generator, its output power can meet the total power demand of all loads. After 0.15s, the synchronous generator output power cannot meet the total power demand of the current task due to the APU taking over the power grid, which requires 35kW load shedding.

For loads with the same power demand, the lower the priority the earlier they are load shed. For example, for loads 4, 5 and 6 with the same power demand, loads 4 and 5 are load shedding before load 6 because their preset priorities are 1, 2 and 3, respectively. For loads with the same priority, the order of load shedding is mainly related to the shedding power demand. For example, loads 3, 4 and 7 all have a preset priority of 1 and a power demand of 3.8, 15 and 11.5 kW, respectively, and when compared with the load shedding power demand, it is known that load shedding load 3 cannot alleviate the overload of the current power generation equipment; after optimization calculations, load shedding is selected for both loads 4 and 7.

Setting the simulation time to 0.3 s, the single-layer and layered control give the same optimized control results when the sampling time of the single-layer control is Ts = 3 × 10-5s. However, the average computation time for the single-layer load control structure but the single-layer simulation is 8.35 minutes, while the average time for the latter is 23.41 seconds. The optimization layer is clocked slower than Ts and hence is able to reduce the amount of computation and the number of iterations of the optimization process, which effectively saves the computation time. It can be seen that the proposed architecture shows the ability to improve computational efficiency.

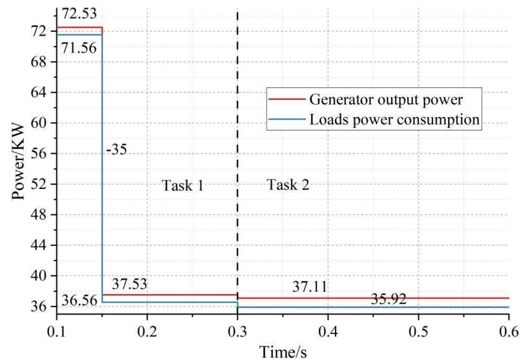

Figure 7: The output power and power consumption of the load in Example 2

IV. Conclusion

The multilevel genetic algorithm (MGA) proposed in this paper effectively solves the multilevel collaborative optimization problem of the electrical drive system of industrial robots, which is embodied in

Algorithm performance superiority: in ZDT1/3/4 standard test, MGA compared with the traditional GA hypervolume (HV) is improved by 4.9% on average, up to 6.2%; the proportion of independent solutions is improved by 5.4 percentage points, such as ZDT3 from 56.21% to 60.68%; the generation distance (GD) and inverse generation distance (IGD) are reduced by a maximum of 72%, and the IGD of ZDT4 from 0.0304 to 0.0084, verifying its strong adaptability to complex nonlinear problems.

In the drive system layout optimization, the convergence speed of MGA is improved by 2.7 times compared with GA, and the objective function value is 4.394 vs. 4.311 for GA. In the dynamic load management experiments, the system achieves an energy efficiency of 98.66% under the healthy working condition, and the main generator failure achieves a precise load shedding of 35kW (48% of the demanded power) through the priority strategy, and the hierarchical control compresses the computation time from 8.35 minutes to 23.41 seconds, which improves the efficiency. The hierarchical control compresses the computation time from 8.35 minutes to 23.41 seconds, improving efficiency by 21 times.

References

[1] Ge, L., Chen, J., Li, R., & Liang, P. (2017). Optimization design of drive system for industrial robots based on dynamic performance. Industrial Robot: An International Journal, 44(6), 765-775.

[2] Quarta, D., Pogliani, M., Polino, M., Maggi, F., Zanchettin, A. M., & Zanero, S. (2017, May). An experimental security analysis of an industrial robot controller. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 268-286). IEEE.

[3] Gadaleta, M., Pellicciari, M., & Berselli, G. (2019). Optimization of the energy consumption of industrial robots for automatic code generation. Robotics and Computer-Integrated Manufacturing, 57, 452-464.

[4] Carabin, G., Wehrle, E., & Vidoni, R. (2017). A review on energy-saving optimization methods for robotic and automatic systems. Robotics, 6(4), 39.

[5] Lü, X., Wu, Y., Lian, J., Zhang, Y., Chen, C., Wang, P., & Meng, L. (2020). Energy management of hybrid electric vehicles: A review of energy optimization of fuel cell hybrid power system based on genetic algorithm. Energy Conversion and Management, 205, 112474.

[6] Rodriguez, J., Garcia, C., Mora, A., Flores-Bahamonde, F., Acuna, P., Novak, M., ... & Aguilera, R. P. (2021). Latest advances of model predictive control in electrical drives—Part I: Basic concepts and advanced strategies. IEEE Transactions on Power Electronics, 37(4), 3927-3942.

[7] Junejo, A. K., Xu, W., Mu, C., Ismail, M. M., & Liu, Y. (2020). Adaptive speed control of PMSM drive system based a new sliding-mode reaching law. IEEE Transactions on Power Electronics, 35(11), 12110-12121.

[8] Cao, F. (2020). PID controller optimized by genetic algorithm for direct-drive servo system. Neural Computing and Applications, 32(1), 23-30.

[9] Zemliak, A. (2022). A modified genetic algorithm for system optimization. COMPEL-The international journal for computation and mathematics in electrical and electronic engineering, 41(1), 499-516.

[10] Min, D., Song, Z., Chen, H., Wang, T., & Zhang, T. (2022). Genetic algorithm optimized neural network based fuel cell hybrid electric vehicle energy management strategy under start-stop condition. Applied Energy, 306, 118036.

[11] Alegre, S., Míguez, J. V., & Carpio, J. (2017). Modelling of electric and parallel-hybrid electric vehicle using Matlab/Simulink environment and planning of charging stations through a geographic information system and genetic algorithms. Renewable and Sustainable Energy Reviews, 74, 1020-1027.

[12] Mallik, S., Mallik, K., Barman, A., Maiti, D., Biswas, S. K., Deb, N. K., & Basu, S. (2017). Efficiency and cost optimized design of an induction motor using genetic algorithm. IEEE Transactions on Industrial Electronics, 64(12), 9854-9863.

[13] Wang, C., Liu, R., & Tang, A. (2022). Energy management strategy of hybrid energy storage system for electric vehicles based on genetic algorithm optimization and temperature effect. Journal of Energy Storage, 51, 104314.

[14] Zhang, L., Liu, W., & Qi, B. (2019). Innovation design and optimization management of a new drive system for plug-in hybrid electric vehicles. Energy, 186, 115823.