ENERGY SAVING OPTIMIZATION OF MACHINING CENTER PROCESS ROUTE BASED ON IMPROVED GENETIC ALGORITHM

Bin LI [1,a]

ABSTRACT: The purpose is to study the energy saving optimization of machining center process route based on the improved genetic algorithm. In modern manufacturing industry, the correct selection of cutting parameters is very important to ensure product quality, improve productivity and reduce production costs. Based on the improved genetic algorithm, the optimization of cutting parameters is studied. By using C++Builder and SQLSever2000 programming, a cutting parameter integration optimization system based on improved genetic algorithm is developed. Through the optimization system, the parameters of the specific machining examples are optimized. The results show that the optimization system of cutting parameters is feasible and universal. Therefore, it can be concluded that the system meets the mechanical process requirements, and improve the processing efficiency.

KEY WORDS: cutting parameters, the improved genetic algorithm, optimization system, machining, process.

1 INTRODUCTION

In machining, the correct and reasonable choice of cutting parameters plays a very important role in ensuring product quality, improving productivity and reducing production costs. It is difficult to guarantee the machining accuracy and machining cost because of the improper selection of cutting parameters. The excessive cutting force and excessive cutting power will cause the machine to stop, thus affecting the normal performance of machine tools performance and efficiency. The selection of optimal cutting parameters in cutting has become one of the most important economic problems in the machine building industry. The continuous change and innovation of machining will also bring great benefits to the national economy. The cutting parameter optimization system aims at improving cutting efficiency, reducing machining cost and obtaining high quality products. It is of great practical significance to study the optimization of cutting parameters.

At present, in the production, most factories are based on experience to select the cutting parameters. It often fails to optimize the cutting parameters [3]. The selection of cutting parameters is an important aspect of the processing technology of machine parts, which will directly affect the quality, productivity and processing cost. In addition, there are many factors that affect cutting parameters [4]. The influence factors are interrelated and restricted. Thus, it is difficult to determine the optimal cutting parameters. A variety of new processing materials and processing technology continue to emerge [1]. With the extensive application of CNC machining tools, CAPP and CIMS, the number of processing is more and more, and new requirements are emerging. The traditional method of cutting parameter determination is far from suitable for the development of machining [2]. For cutting, milling and other cutting methods, the cutting parameters are integrated and optimized. The hybrid genetic algorithm is used as an optimization tool to optimize the parameters. The feasibility of the integrated optimization system is verified by a specific example.

2 MATERIALS AND METHODS

2.1 The introduction of genetic algorithm

Genetic algorithm (GA) is a global search algorithm, which randomly generates a population. Then. it simulates the process of natural evolution in a better direction, and evaluates the merits of individuals through fitness. In each generation, the superior individual produces offspring by genetic manipulation, and the inferior individual is eliminated. Individuals in a population are usually composed of a string of binary or real numbers of length. An individual is a solution to the problem, and genetic algorithms are widely used to solve optimization problems.

The genetic algorithm was first proposed by American professor John. Holland of the University

[1] Xinxiang Vocational and Technical College, Henan, China

E-mail: [a]technical_binli@yeah.net

of michigan. It originated from the study of adaptive behavior of natural and artificial systems in the 60s. It is a global optimization random search algorithm, which imitates the principles of natural biology, heredity and evolution. It uses bit string coding technique to generate initial population for the problem. Then, the fitness of the population is evaluated, followed by selection, crossover, and mutation operations. A selection strategy based on fitness ratio is used to select individuals in the current population, and crossover and mutation are used to produce the next generation. This generation goes on until it meets the expected conditions. The genetic algorithm can simultaneously search multiple regions in the solution space by using the method of population organization, and it is especially suitable for large-scale parallel processing. Genetic algorithm has the characteristics of self-organization, self-adaptation and self-learning. The natural selection of the fittest and the simple genetic operations make the computation free from the constraints of the search space (e.g., differentiable, continuous, unimodal) and the absence of additional supporting information (such as guidance). Therefore, the genetic algorithm not only can achieve high efficiency, but also has a simple, easy to operate and common features. At present, with the development of computer technology, genetic algorithm has been paid more and more attention, and has been successfully applied in machine learning, pattern recognition, image processing, combinatorial optimization, VLSI design and optimization control. Genetic algorithm is not a simple random comparison search algorithm. By assessing the fitness of chromosomes and the role of genes in chromosomes, it effectively uses existing information to guide the search for the most promising states to improve the quality of the optimization.

The basic genetic algorithm can be defined as an 8-tuple:

$$GA = (C,\ E,\ P_0,\ M,\ \phi,\ \chi,\ \partial,\ T)$$

In the above formula: C is the coding method of the individual, E is the individual fitness evaluation function, P0 is the initial population operation, M is the population size, ϕ is the selection operator, χ is the operator, ∂ is the mutation operator, T is the convergence operator of the genetic operation.

The improvement of genetic algorithm

The genetic algorithm emphasizes the evolutionary relationship between the two generations. It has a strong global search capability and can take advantage of the knowledge in the previous search process [5]. However, the local search ability is poor. It is easy to fall into the local optimal solution, and the convergence speed is slow. The simulated annealing algorithm has a strong local search capability. It can get a better approximation solution and avoid falling into local optimum. However, it is difficult to determine which regions in the space have more opportunities to get the optimal solution, and the algorithm takes a longer time. Therefore, combining the advantages of the two algorithms, it is feasible to integrate the simulated annealing algorithm into the genetic algorithm [6].

The simulated annealing genetic algorithm (SAGA) was originally introduced by Paul L.Stoffa, which combines the simulated annealing algorithm with the genetic algorithm. The combination method is to insert the simulated annealing algorithm into the crossover and selection part of the genetic algorithm. Under this algorithm, the genetic algorithm is responsible for global search in the population, and the simulated annealing algorithm is used for the local search of chromosomes [8].

The simulated annealing genetic algorithm is improved as follows. In order to solve the premature convergence problem in the genetic algorithm, the selection mechanism in the simulated annealing algorithm is introduced into the genetic algorithm. This preserves the diversity of the population and avoids the problem of poor convergence [9]. The selection mechanism takes the Metropolis acceptance criterion as the evaluation criterion, so that the individual in the group is gradually optimized to the optimal solution.

The specific operation is as following. The temperature is t. a new state j is generated from the current i state. The fitness functions f (i) and f (j) of state t and state j are calculated according to equations (1) and (2). If f(i)≤f(j), the new state j replaces state i and becomes the current state. If f(i)＞f(j), then compare the Metropolis acceptance probability $P_k = \exp(\frac{f(i)-f(j)}{t})$ with the random number Prob within [0,1], at this moment, if f(i)≤f(j), Pk＞Prob, the new state j replaces state i and becomes the current state. Otherwise, the former state is still i.

The algorithm uses the following fitness stretching method:

$$f_i = \frac{e^{f_i/T}}{\sum_{i=1}^{M} e^{f_i/T}} \qquad (1)$$

$$T = T_0(0.99^{g-1}) \quad (2)$$

In the formulas (1) and (2), f (i) is the fitness of the i-th individual, M is the population size, g is the genetic algebra, T is the temperature, and T0 is the initial temperature. The genetic algorithms are relatively large in the early stages of operation. Therefore, it is easy to make the individual good individual offspring filled the whole population, leading to premature. In the later period, the fitness tends to be consistent, so that the evolution of the whole population is stagnant. Therefore, it is necessary to properly stretch the fitness. At high temperatures, individuals with similar fitness have similar probability of offspring. When the temperature decreases, the stretching effect is strengthened, and the difference of the fitness is close to the individual, so that the advantage of the outstanding individual is more obvious. The improved hybrid genetic algorithm flow chart is as shown in Figure 1.

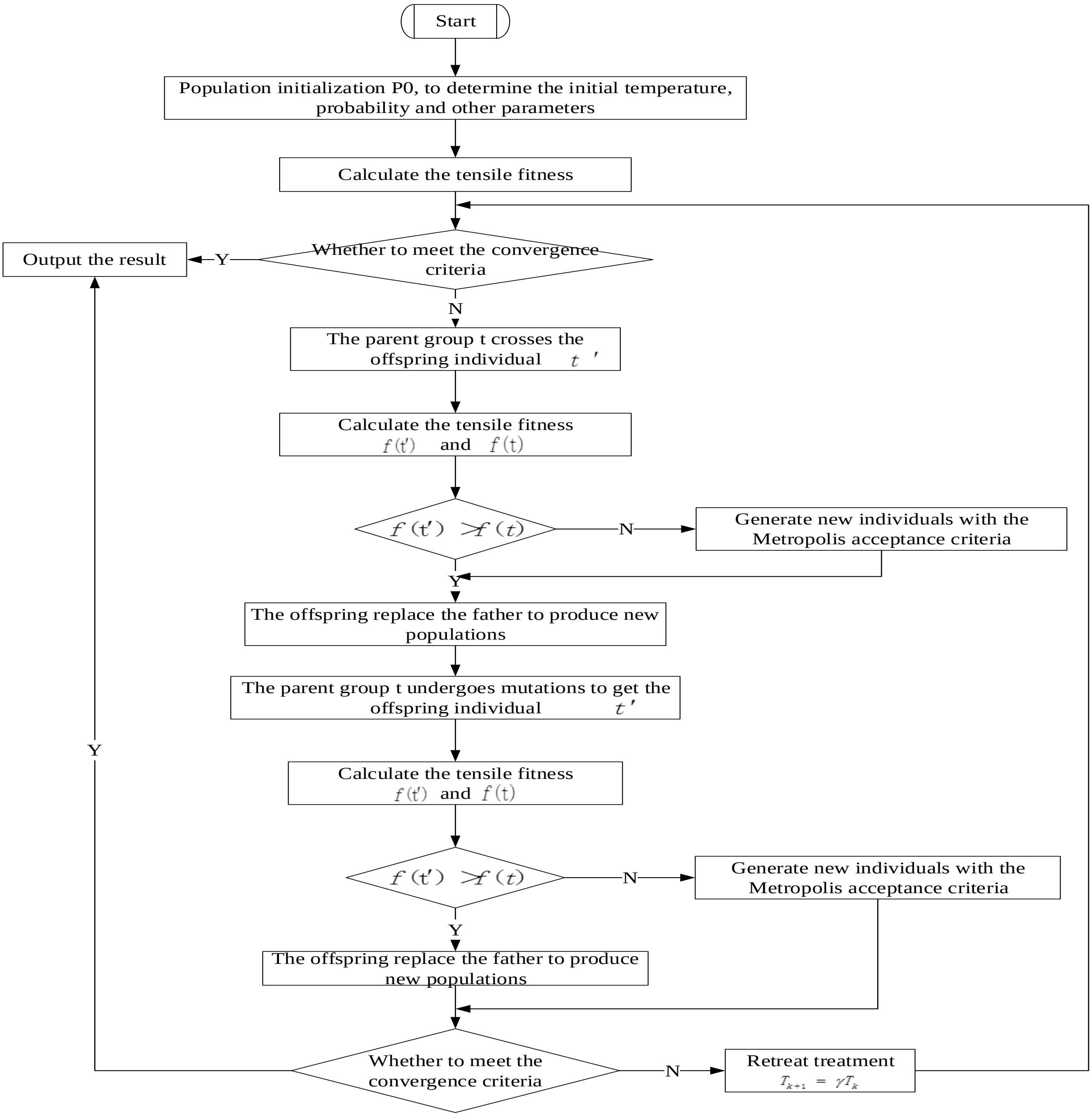

Figure 1. The improved flow chart of hybrid genetic algorithm

2.2 Algorithm performance test

In order to verify the efficiency and convergence of the improved genetic algorithm, the commonly used De Jong's F2 function of the genetic algorithm is used to compare the actual performance of the two algorithms [7]. De Jong's F2 function is:

$$f(x,y) = 100(x^2 - y^2) + (1 - x)^2 \qquad (3)$$

Among them, $x \in (-2,2), y \in (-2,2)$. De Jong's F2 function is a two-dimensional function with a minimum point f (1.0, 1.0) = 0. However, the minimum value of this function has a narrow valley around the ridge. The traditional gradient optimization method is easy to fall into the vicinity of the local extreme point. As a result, it is difficult to get the minimum point. For the genetic algorithm and the improved algorithm, the coding method adopts the real number coding, which avoids the mapping errors of the binary code and reduces the complexity of the genetic algorithm. The fitness function of De Jong's F2 function is set to $f_{fitness}$ = 3700-f (x, y), the crossover probability is set to 0.8, and the mutation probability is 0.1.

For the initial population size and the operation of algebra settings, due to the particularity of De Jong's F2 function, it is very easy to fall near the local extreme point, which is difficult to get the minimum point. When the initial population size is 500 and the run algebra is 500, the result of running the algorithm is shown in Table 1.

Table 1. The result of GA when initial population number is 500

Operational algebra	Adaptive value
1	3999.88304
2-12	3699.96306
13	3699.97915
14	3699.98502
15-57	3699.99510
58-500	3699.99998

It can be seen that the fitness function is easy to fall into the extreme point (1.004, 1.008), and the fitness function value is 3699.99998. It cannot reach the optimal result. Therefore, the initial population size is 1000, and the operation algebra is 500. The genetic algorithm and the improved hybrid genetic algorithm are used to solve the fitness, maximum algebra and individual of De, Jong, s and F2 functions respectively. After running, the results of the two algorithms are shown in Table 2 and table 3, respectively:

Table 2. The result of standard genetic algorithm verification

Operational algebra	Adaptive value
1-2	3699.96054
3	3699.96979
4-67	3699.99828
68-71	3699.99837
72-177	3699.99998
178-500	3700.00000

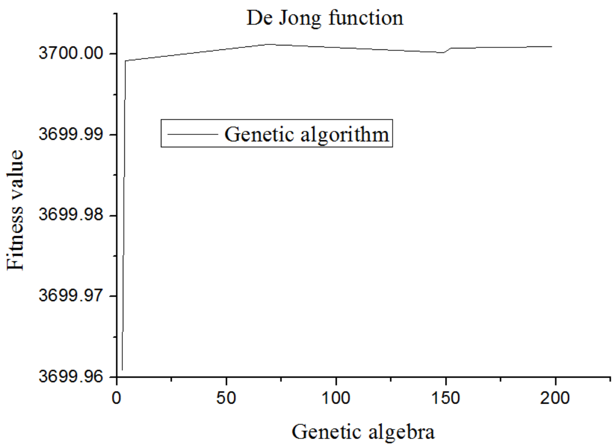

Figure 2. The operational efficiency of standard genetic algorithm

Table 3. The verification results of the improved algorithm

Operational algebra	Adaptive value
1-2	3699.96164
3-6	3699.96871
7-10	3699.99828
11-15	3699.99845
16-17	3699.99992
18-20	3699.99998
21-500	3700.00000

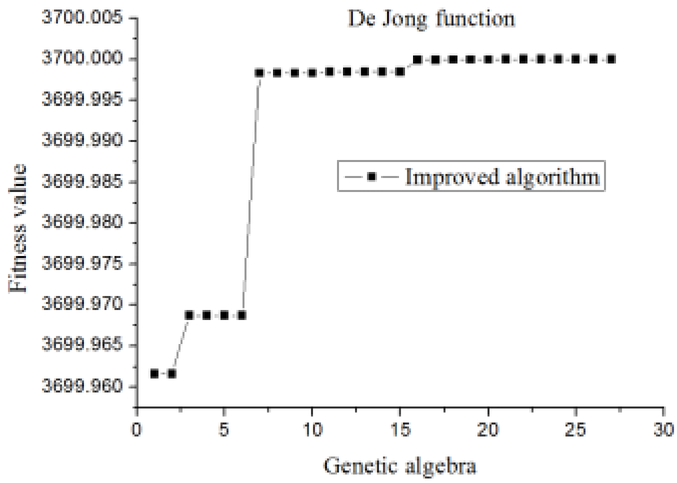

Figure 3. The efficiency of the improved algorithm

Through the comparison of the two algorithms, it can be concluded that the standard genetic algorithm obtains the optimal individuals in 177 generations. The improved hybrid genetic algorithm achieves the best fitness in the 21st generation. Therefore, the local search ability of standard genetic algorithm is poor and the efficiency is low. It is easy to fall into extreme points (1.004,1.008). The population number is set to 1000, and the optimum value appears for 177 generations. If the population number is set to 500, it is caught in the local extremum, and cannot get the optimal solution. The improved hybrid genetic algorithm only runs the 21th generation to get the optimal individual and does not fall into the extreme value. Therefore, compared with the traditional genetic algorithm, the improved hybrid genetic algorithm is more efficient and faster. It not only has the traditional search ability of traditional genetic algorithm, but also has a strong local search ability of simulated annealing algorithm. In addition, it also improves the convergence of traditional genetic algorithms at a later stage. Therefore, the combination of simulated annealing algorithm is effective for the improvement of traditional genetic algorithm. The correctness of the improved algorithm is verified by De Jong's F2 function, which provides a more efficient algorithm for parameter optimization.

3 RESULTS

3.1 The development environment of the system software

The development environment of the system software is: Hardware environment: CPU Pentium (R) D 2.66 GHz, memory 512MB, graphics card NVIDIA GeForce7300LE128MB and hard disk 128GB. Software environment: Operating system Windows XP, development tools Borland C ++ Builder 6.0, database SQL Server 2000 and operating environment Pro / E Wildlife 3.0.

3.2 Process and module of software system

By analyzing the functional requirements of the cutting parameter optimization software system, the process of cutting parameter optimization can be derived. The specific flow chart is shown in Figure 4:

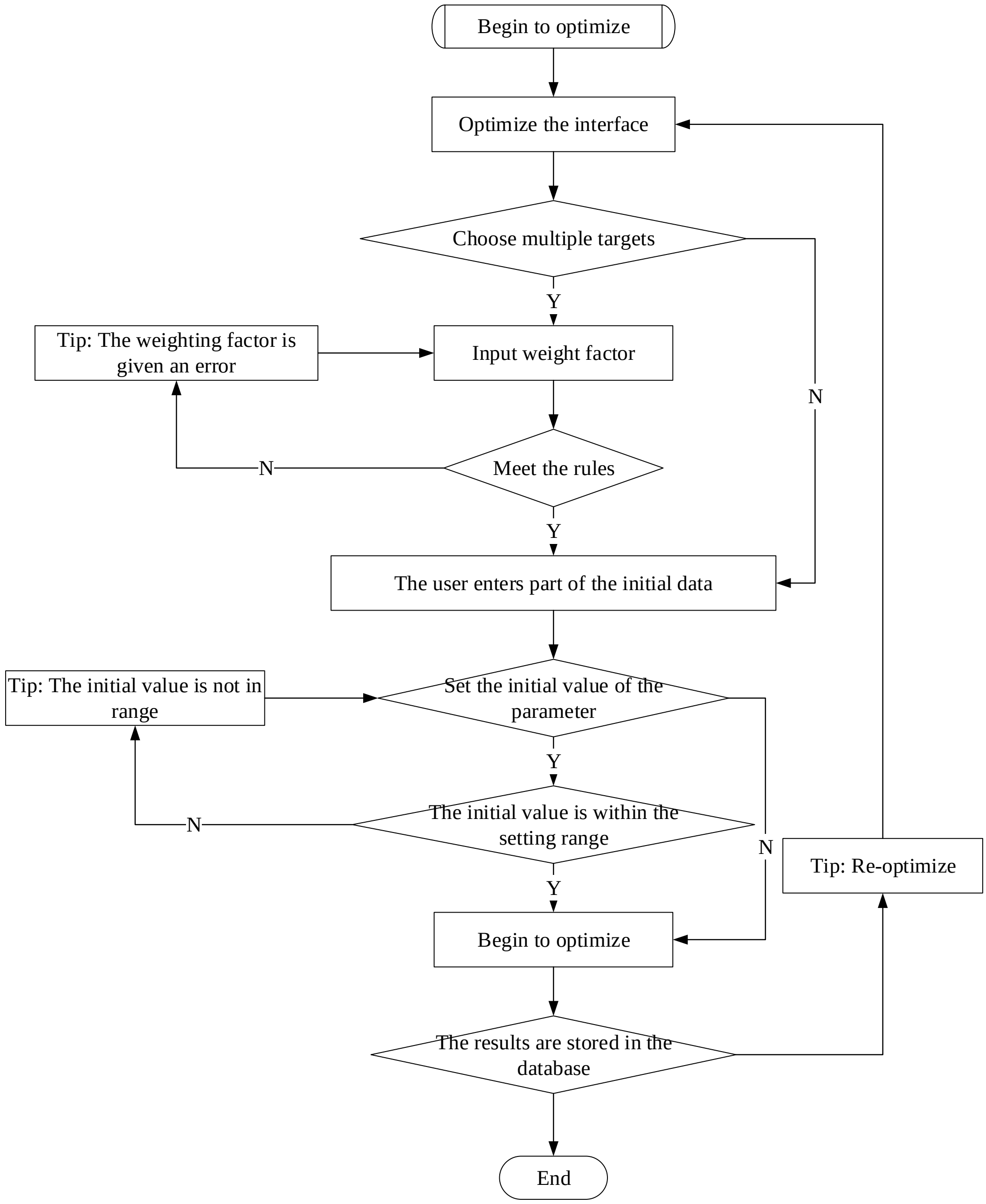

Figure 4. The flow chart of system optimization of cutting parameters

3.3 Instance validation

In order to verify the feasibility of the integrated optimization of several cutting parameters in this paper, an experiment is carried out in a manufacturing enterprise. The box parts are shown in Figure 5. It involves a variety of cutting methods. The box is produced in large quantities, and the casting blank with a single margin of 4mm is used.

Figure 5. Box parts drawing

The process personnel hope that the cost is low, production profit is higher. The weight coefficient is set as follows: production time is 0.4, and the production cost is 0.6. The parameters of genetic algorithm are set as follows: population size Npop=20, iteration number 500, crossover probability Pc=0.8, and the variation probability Pm=0.05. For the process that requires parameter optimization, the optimized results are shown in Table 4.

Processing order	Feature processing method	Machine tool	Cutter	Cutting parameters		
				Cutting depth (mm)	Feed (mm / r)	Cutting speed (m / min)
1	Top line alignment					
2	Box joint surface	C5112A	YG6	6	0.5	70
3	Half-finished turning upper case joint surface	C5112A	YG6	1	0.3	107
4	Finished turning upper case joint surface	C5112A	YG6	0.5	0.15	130
5	8 countersunk heads on the countersink	Z3050	Tapered countersink	2	0.1	20
6	8 holes in the joint surface	Z3050	High-speed steel drill	25	0.27	13
7	lower housing lineation					
8	Rough milling surface	TK6111	Butt mill	3	0.28	55
9	Junction surface of rough lower case	TK6111	YG8	3	0.28 (mm/z)	57
10	Half-finished turning lower case joint surface	TK6111	YG8	1	0.1 (mm/z)	85
11	Finished turning lower case joint surface	TK6111	YG8	0.5	0.05 (mm/z)	92
12	Countersink box on the 8 countersunk heads	Z3050	Tapered countersink	2	0.1	20
13	Drill off 8 light holes on the cabinet	Z3050	High speed steel drill bit	25	0.27	13
14	Countersunk counters on the four countersunk heads	Z3050	Tapered countersink	2	0.1	20
15	4 light holes on the drill base	Z3050	High speed steel drill bit	20	0.39	15
16	Mould assembling					
17	Rough boring is located on the 6 spindle holes on the joint surface	TK6111	YT5	5	0.6	51
18	The 6 main spindle holes on the joint face	TK6111	YT30 Fine boring knife	1	0.28	63
19	Milling the window end face	TK6111	YG8	1	0.1	85
20	Drill the 6 holes on the end face of the window, tapping	Z3050	High speed steel drill bit	15	0.21	12
21	Drill two holes on the boss, tapping	Z3050	High speed steel drill bit	25	0.27	13

In the optimization of turning, milling and boring, the cutting force empirical formula and the material removal power can be used in terms of processing constraints. In the processing of countersink, drilling and other processing methods of optimization, the study of the experience of cutting force is not enough depth. Therefore, the cutting power constraint in terms of material removal rate is used.

The evaluation of the test results must take into account the cost of production and cutting time, as well as the effectiveness of the overall system.

Therefore, the results of the optimization and the results of the machining process manual can be considered from the two aspects of the comprehensive consideration. The milling case of the finishing process 9 is taken as an example. The results are shown in Table 5.

	Cutting depth (mm)	Feed (mm / r)	Cutting speed (m / min)	Cutting force (N)	Power (kw)	Ra	Processing time (s)
Optimization value	0.5	0.05	92	2800	7.5	1.6	202
Recommended values for the manual	0.5	0.05	80	1900	7.5	1.6	270

4 CONCLUSION

Through the experiment, the results show that the system meets the mechanical processing requirements, and greatly improve the efficiency. The optimized results were confirmed to be the best results. The processing of the parts in Figure 3 illustrates that the system meets the requirements for processing costs and processing time. In the processing mode, it is suitable for boring, milling, car, drilling and other different ways. The machining quality of the parts is guaranteed by the parameter results. Therefore, the availability of the optimization results is relatively high. It proves the versatility and practicability of the integrated optimization program.

5 REFERENCES

►Sahali, M. A., Belaidi, I., & Serra, R. (2015). Efficient genetic algorithm for multi-objective robust optimization of machining parameters with taking into account uncertainties. International Journal of Advanced Manufacturing Technology, 77(1-4), 677-688. DOI: 10.1007/s00170-014-6441-z

►Varun, A., & Venkaiah, N. (2015). Simultaneous optimization of wedm responses using grey relational analysis coupled with genetic algorithm while machining en 353. International Journal of Advanced Manufacturing Technology, 76(1-4), 675-690. DOI: 10.1007/s00170-014-6198-4

►Li, J., Yang, X., Ren, C., Chen, G., & Wang, Y. (2015). Multiobjective optimization of cutting parameters in ti-6al-4v milling process using nondominated sorting genetic algorithm-ii. International Journal of Advanced Manufacturing Technology, 76(5-8), 941-953. DOI: 10.1007/s00170-014-6311-8

►Wu, M., Huo, T., & Ge, J. (2016). Cutting process-based optimization model of machining feature for cloud manufacturing. International Journal of Advanced Manufacturing Technology, 84(1), 1-8. DOI: 10.1007/s00170-015-7800-0

►Ming, W., Ma, J., Zhang, Z., Huang, H., Shen, D., & Zhang, G., et al. (2016). Soft computing models and intelligent optimization system in electro-discharge machining of sic/al composites. International Journal of Advanced Manufacturing Technology, 1-17. DOI: 10.1007/s00170-016-8455-1

►Santhanakrishnan, M., Sivasakthivel, P. S., & Sudhakaran, R. (2015). Modeling of geometrical and machining parameters on temperature rise while machining al 6351 using response surface methodology and genetic algorithm. Journal of the Brazilian Society of Mechanical Sciences & Engineering, 1-10. DOI: 10.1007/s40430-015-0378-5

►R. Venkata Rao, Dhiraj P. Rai, & J. Balic. (2016). Multi-objective optimization of machining and micro-machining processes using non-dominated sorting teaching–learning-based optimization algorithm. Journal of Intelligent Manufacturing, 1-23. DOI: 10.1007/s10845-016-1210-5

►Kuriachen, B., Somashekhar, K. P., & Mathew, J. (2015). Multiresponse optimization of micro-wire electrical discharge machining process. International Journal of Advanced Manufacturing Technology, 76(1-4), 91-104. DOI: 10.1007/s00170-014-6005-2

►Kara, F., Aslantas, K., & Çiçek, A. (2015). Ann and multiple regression method-based modelling of cutting forces in orthogonal machining of aisi 316l stainless steel. Neural Computing & Applications, 26(1), 237-250. DOI: 10.1007/s00521-014-1721-y