APPLICATION OF GENETIC PARTICLE SWARM OPTIMIZATION ALGORITHM IN BEARING MANUFACTURING WORKSHOP

Wang Yahui[1], Xing Shaoqun[1], Jiang Kaixin[1], Fu Liuqiang[1], Tian Xiaoguang[2]
[1]*School of Mechanical Engineering, North China University of Water Resources and Electrical Power Zhengzhou, Henan 450045, China*
[2]*College of Mechanic and Electronic Engineering, Huang He Jiao Tong University, Jiaozuo, Henan, 454950, China*
Email: wangyahui@ncwu.edu.cn

ABSTRACT: *In multi-objective optimization problem, according to the regularity of Pareto solution set distribution, the algorithm of multi-objective flexible production scheduling in structured dynamic environment is studied. In the process of designing the algorithm, the inherent structure of the problem should be fully considered so that the algorithm can make full use of the structural information of the problem in the process of searching. According to the established multi-objective scheduling model with workpiece machining cycle, advance or delay penalty and processing cost as the objective function, using the command window, editor and debugger modules in MATLAB tool, combining the advantages of genetic algorithm and particle swarm optimization, the genetic algorithm operation is carried out by using the population after particle swarm optimization to output Pareto optimal solution set.*
Keywords*: Multi-objective optimization, Genetic algorithm, Particle swarm optimization*

1 INTRODUCTION

Job shop scheduling is an important part of workshop production management in manufacturing industry. When the manufacturing industry is facing a complex market environment Constrained by production resources, if it can make scientific decision of production scheduling reasonably and respond to market demand rapidly, it can not only improve the utilization rate of equipment, reduce the manufacturing cycle of products, but also improve product quality and reduce production costs (Luh P B, 1990). Therefore, how to use computer technology to optimize production scheduling plan, adjust resource quickly allocation, coordinate production schedule and improve equipment utilization rate has become a major issue for many processing enterprises (Wang Yu ping, 2011).

2 DESCRIPTION AND MODEL CONSTRUCTION OF MULTI-OBJECTIVE FLEXIBLE JOB SHOP SCHEDULING PROBLEM

2.1 Problem description

Processing a batch of parts in a workshop, the number is n (n=1, 2, 3…N). The machine tool in the workshop can process the workpiece. The number of machine tools is k (k=1, 2, 3…k), and the number of processes is m (m=1, 2, 3…m). Each machine can only produce one process of a workpiece at the same time, and the same or different processes of subsequent workpieces can be produced only after the completion of processing. For machine tools, because the front workpiece is being processed by this machine, the workpiece process and machine tool is both occupancy constraints; the latter workpiece and machine tool are sequential waiting constraints. (Berce, Petru; Pacurar, Razvan; Balc, Nicolae. 2008).

Assuming that the machine is in the state to be processed at t=0 and the machining objects are processed in accordance with the established machining plan; the flexible FJSP is divided into the following two categories:

Fully flexible FJSP: For V_{yz} , $\left|KF_{yz}\right| = x$, Processing conditions of processes are not limited by machines, and machines in non-occupied state can meet their operating conditions.

Partially flexible FJSP: For V_{yz} , $\left|KF_{yz}\right| < x$, Processing conditions of processes are limited by machines. Even if the machines in the arrangement are idle, the machines can only wait or reschedule because they do not meet the performance requirements.

2.2 Construction of optimized mathematical model

2.2.1 Construction of problem model

(1) The objective function f_1 is established with the shortest processing time of the workpiece.

$$\min f_1 = \sum_{o=1}^{O} \sum_{m=1}^{M} \sum_{n=1}^{N} (T_{omn} - t_{omn}) \quad (1)$$

(2) Based on the punctuality of delivery date, the objective function f_2 is established to minimize penalties for delay or early completion.

$$\min f_2 = \sum_{m=1}^{M} r \times \max(0, k_{omn} - d_m) \quad (r = r_{m1} \; or \; r = r_{m2}) \quad (2)$$

(3) Establishing objective function f_3 with minimum processing cost.

$$\min f_3 = \sum_{o=1}^{O} \sum_{m=1}^{M} \sum_{n=1}^{N} U_{omn} + \sum_{m=1}^{M} r \times \max(0, k_{omn} - d_m) + u \quad (3)$$

"m" is the serial number of the workpiece, m=1,2,3…M. "n" is the serial number of the process, n=1,2,3…N. "o" is the corresponding machine serial number, k_{omn} represents the process "n" of workpiece "m" processed on machine "o", U_{omn} represents the production cost of workpiece "m" processed on machine "o", "U" represents other costs, t_{omn} represents the starting time of workpiece "m" processed on machine "o", t_n is the processing time at the completion of process "n".

T_{omn} represents the processing completion time of process "n" of workpiece "m" on machine "o". d_m represents the delivery date of the workpiece "m", "D" represents the actual delivery date of the workpiece "m", r_{m1} is the penalty factor for the delay completion of the workpiece "m", r_{m2} is the penalty factor for the pre-completion of the workpiece "m".

2.2.2 Constraint condition

$$\begin{cases} D < d_m & r = r_{m2} \\ D \geq d_m & r = r_{m1} \end{cases} \quad (4)$$

$$\begin{cases} k_{omn} - k_{o(m-1)n} - o_{omn} \geq 0 \\ x_{omn} = x_{o(m-1)n} \\ 1 < m \leq M \end{cases} \quad (5)$$

$$\begin{cases} k_{omn} - k_{oij} - o_{om} \geq 0 \\ x_{omn} = x_{o1ij} \\ t_{omn} > t_{oij} \end{cases} \quad (6)$$

$$\begin{cases} T_{o(m+1)n} \geq t_{omn} + T \\ T_{omn} \geq t_{omn} \\ t_n > 0 \end{cases} \quad (7)$$

When the workpiece is processed on the machine, the operation of the next process can only be carried out after one process of the workpiece is processed by the machine.

After the completion of the workpiece process, the processing time is sequential and the processing time of the same process is fixed, which involves the waiting time of the machine stopping, and the working time cannot be zero in the time setting.

3 DESIGN FLOW OF DIFFERENT ALGORITHMS FOR JOB SHOP SCHEDULING PROBLEM

3.1 Design flow of genetic algorithm for solving FJSP

The genetic algorithm takes the biological evolution mechanism as the research direction, and has obtained many mature and stable theoretical results in decades of research. It has laid a solid theoretical foundation for practical application. Selecting genetic algorithm to solve multi-objective flexible job shop scheduling (Gen, M., 2000) can not only make use of the good solving ability and optimization efficiency of genetic algorithm, but also deepen the understanding of the algorithm in future work and study.

As shown in Figure 1, the scheduling executor will formulate the production scheduling rules according to the necessary information that the production task allocation conforms to the processing production, and select the appropriate scheduling evaluation function according to the corresponding production order. The numerical survival of the fittest is carried out through the continuous cross-recombination of genetic operators and the optimal results are displayed. The design of genetic operation (crossover, selection, mutation) is very important to the performance of the algorithm. The job shop scheduling problem is different from the general numerical optimization problem. It is applied to the solution and design of practical schemes, and it is also NP-hard. In operator solving, both practical feasibility and production cost should be taken into account, and if the cost is relatively large, a good scheduling plan will lose its usefulness.

3.1.1 Coding and decoding design

In this paper, the coding calculation based on priority rule coding is carried out. For the representation of this coding, chromosome is a gene string $u_1, u_2, \ldots, u_{mn}$ containing $n \times m$ genes. In order to avoid falling into local optimum in operation and conflict in operator reorganization operation, each gene string corresponds to a priority rule according to the coding arrangement (Lei Deming, 2006). Based on the existing scheduling methods, the traditional $n \times m$ FJSP scheduling problem is expressed as a chromosome constrained by the relationship, which is more in line with the existing problem model. The expression of chromosomes containing $n \times m$ genes is as follows :($u_{11}, u_{12}, \ldots, u_{1z}, \ldots, u_{mn}$).In the encoding scheduling information, chromosomes are constrained by relationships, such as u_{11} representing the first n-bit relational machine 1, and then the relationship constraints are applied to machine 2, and the machines are sequentially correlated in this order. The same as the traditional $n \times m$ FJSP scheduling problem, this method also has the corresponding priority rules in use. Decoding using the GT process is as follows:

(1)Setting the starting time $t = 1$, PS_t is an empty set with no spatial value, S_t is an executable scheduling set with time "t".

(2)Because of the corresponding relationship, the choice of processing and process may be processed on more than one machine, and then a pair of random matches can be selected.

(3)A mutually exclusive set $c_t = \{o_{im}^* \in S_t \,|\, \sigma_{im}^* < C(o^*)\}$ is established, and an operable process is selected from the mutually exclusive set by using the set conditions to operate, and $PS_{t+1} = PS_t \cup \{o_{im}^*\}$ is obtained, and o_{im}^* is removed from the original set S_t so that the executable follow-up steps of the workpiece can be summarized into S_t, that is, S_{t+1}. ($C(o_{ij})$ is the earliest processing completion time, σ_{ij} is the earliest processing start time)

(4)Let t = t + 1 go to step (2) to continue the operation and become the complete scheduling of a new cycle. In operator calculation, in order to make gene encoding search efficiently, the constraints matrix designed should be easy to implement in algorithm operation, and the selection of priority rules should be combined with objective function to solve constraints for different types of conditions.

3.1.2 Selection method of genetic algorithms

The genetic algorithm usually uses roulette selection method to operate the algorithm. Whether the descendants are close to or alienated from the fitness value, the probability of the descendants participating in the next generation operator reorganization is same. In order to ensure that the fitness value of the optimal solution does not decrease due to operator reorganization, the elite retention strategy is adopted to replace the worse one without operator reorganization. In order to ensure the diversity of Pareto solutions in each generation of evolution, niche technology is used to deal with the Pareto optimal solution.

The operation process of niche technology is to arrange individuals in descending order according to fitness and reorganize operators of the first N individuals, to select, cross and mutate the generated population, to calculate penalty function using niche elimination algorithm and to select a new population in descending order, and to operate repeatedly until the optimal outcome is achieved. A design method is proposed by Hyu (HYUN C J, 1998) in the literature, which is centered on the individual. The calculation of the periphery boundary is shown in equation (8). It can be seen that the density of niche is inversely correlated with the descending probability of individual heredity, and the greater the density of niche, the greater the probability of individual elimination.

$$s_{lt} = \frac{\max f_{it} - \min f_{lt}}{\sqrt[M]{N}} \tag{8}$$

$$s_{lt} = \frac{\max f_{it} - \min f_{lt}}{2\sqrt[M]{N}} \tag{9}$$

Among them, $\max f_{lt}$ and $\min f_{lt}$ denote the maximum value of target l (l=1,2,3…M) when operator reorganization is performed and N denotes population size.

It is found that the number of optimal solutions is too large or too small when using the above-mentioned methods. That is, the higher the density of the habitat, the more the optimal solution that should be retained in the calculation of the new population will be eliminated. The smaller the density of the habitat, the more the optimal solution that should be retained in the evolution will reduce the convergence rate. After several simulation comparisons, the revised formula is shown in formula (9). (Leordean, Dan;

Radu, S. A.; Fratila, D.; et al. 2015)

3.2 Particle coding decoding

In solving flexible job shop scheduling problem by PSO, how to establish the mapping relationship between location vector and scheduling scheme according to the actual production tasks is the key to design PSO to solve FJSP problem (I Kennedy J, 1995; Pan Feng, 2006). Particle positions are defined as two mutual O-dimensional vectors P [Z] and M [Z].Z represents the total number of processes in the production plan. As shown in Table 2-1, the particle coding of a processing workshop shows that the particle is represented by a 2×8 -dimension vector, and the natural number of P[Z] represents the workpiece number. For example, the first process of workpiece 1 is on P [Z] 1 dimension, and machine 2 on M [Z] 1 dimension is used. The second process is on P [Z] 4 dimension, using machine 3 on M [Z] 4 dimension. The particle position vector represents the processing order of the process, which is the priority of scheduling.

Table 1. Particle coding in a workshop

Dimension	1	2	3	4	5	6	7	8
P[Z]	1	3	4	1	2	1	2	4
M[Z]	2	1	4	3	2	1	2	3

3.2.1 Calculation of position and velocity

Each particle is marked as "m", and $m \in \{1,2,3 \cdots M\}$, the position of particle "m" in space is marked as P, $P_m = (p_1, p_2 \cdots p_O)$, "o" represents the spatial dimension, the velocity of particle movement is marked as v, $v = (v_1, v_2 \cdots v_O)$, the optimum position of particle "m" is marked as pbest, and the global optimum is marked as gbest. The position and velocity of particle movement can be expressed by formulas (10) and (11):

$$v_{t+1} = wv_t + r_1 * rand(\ ) * (P_t - m_t) + r_2 * rand(\ ) * (G_t - m_t) \tag{10}$$

$$m_{t+1} = m_t + v_t \tag{11}$$

"w" denotes the inertia coefficient of flight itself, r_1 and r_2 denote the acceleration constant, P_t and M_t denote the pbest and gbest at "t" time, respectively.

According to the difference of inertia coefficient and acceleration constant, the value obtained by weighting the position and velocity of particles when PSO solves FJSP problem may no longer be an integer. Machines and corresponding processes are actually integers in particle encoding. Therefore, the value space of the x-dimensional vector ($0 < x \le O$) is defined as [1, O]. If P [Z] =4.2 is weighted, it is treated as an integer. The results of several iterations in Table 2 are as follows:

The results of P [Z] component were arranged in ascending order: 1.2 < 1.3 < 1.7 < 1.8 < 1.9 < 2.1 < 2.3 < 2.5

The updated particle components correspond as shown in Table 3.

Table 2. After several iterations

Dimension	1	2	3	4	5	6	7	8
P[Z] Before iteration	1	3	4	1	2	1	2	4
P[Z] After iteration	1.8	1.9	2.1	2.3	1.7	2.5	1.3	1.2

Table 3. Particle vector relationship after iteration

Dimension	1	2	3	4	5	6	7	8
P[Z]	1	3	4	1	2	1	2	4
M[Z]	1	1	4	3	2	2	2	3

3.2.2 Weight and fitness selection

After decoding the established model by particle swarm optimization, we can get a set of objective functions that aim at the flow time of the dispatched workpiece in processing, the penalty of production cost, advance or delay of delivery.

In order to aggregate the above data into chromosome fitness, it is necessary to dimensionalize the index values of scheduling scheme according to formula (12) (Coello Coello C A, 2004), that is, to aggregate the values of multiple objective functions into chromosome fitness.

$$a_{ij} = \frac{99 \times \left[b_{ij} - \min_i b_{ij} \right]}{\max b_{ij} - \min_i b_{ij}} + 1 \qquad (12)$$

a_{ij} represents the j-th indicator in the i-th scheme after the dimensioning process;

b_{ij} represents the j-th indicator in the i-th scheme;

$\min_i b_{ij}, \max_i b_{ij}$ respectively represent the minimum and maximum values in the scheme;

After the processing, according to the established shop scheduling model, the workpiece processing time $f_1(x)$, the processing cost $f_4(x)$, the penalty of the advance delivery $f_2(x)$ or the penalty of the backorder $f_3(x)$ are quantized, and “x” represents the individual position vector string of the particle.

3.3 The idea of hybrid genetic particle swarm optimization

The traditional genetic algorithm has better robustness and searching ability in problem optimization, and can grasp the overall evolutionary direction. It is a commonly used optimization method in solving job shop scheduling problems.

However, in the later stage of search, due to the huge data and poor target selectivity, the problem of inefficiency and premature convergence is easy to arise.

PSO algorithm has simple calculation and high efficiency. The selected individuals can directly guide the evolution direction of the next generation of particles. However, due to its weak theoretical basis and short development time, it still has the shortcomings of being easy to premature and falling into local optimum, so it is also constrained by certain conditions when it is used.

Combining the previous study of the theoretical knowledge of the algorithm, the combination of genetic algorithm and PSO algorithm is sought to make up for the deficiencies of the algorithm in operation and to give full play to its advantages. It has important application value for solving multi-objective flexible job shop scheduling problem.

3.4 Implementation flow of hybrid genetic particle swarm optimization

According to Figure 1, the implementation steps of the algorithm are given.

(1) Coding: When coding the shop scheduling information, in order to ensure the connectivity of the information used by the algorithm, the genetic algorithm and the particle swarm algorithm are used to perform the process-based coding method. The process of the same workpiece is assigned the same symbol. The number of times represents the process of the workpiece. The process is processed on the machine and the machine is recorded without encoding. One machine can process multiple processes, and each process can be processed on multiple machines.

(2) Parameter initialization: setting population crossover rate, mutation rate, iteration times, population size and other parameters of genetic operation; setting the number of particle swarm, particle swarm size, acceleration factor, inertia weight coefficient and other parameters.

(3) Generating initial particle swarm: Establishing initial particle swarm according to the coding method of the process. Each group contains multiple particles. According to the corresponding relationship between P [Z] and M [Z], the number of occurrences of each particle corresponding to the process and the machine serial number of processing are selected.

(4) Particle swarm velocity initialization and position initialization: Acceleration constants and inertia weight coefficients are set according to formulas (8) and (9).

(5) Calculating fitness: Fitness function formula (12) is selected for objective function with scheduling model, and its related parameters are determined according to the need. After fitness weighting calculation, the particles will produce good individuals and worse individuals. After particle search, Pbest and Gbest are selected according to the space vectors P [Z] and M [Z] of the particles.

(6) Sorting the particles: According to the descending order of the optimized particles, the second half of the cases with low fitness are

selected for selection, crossover and mutation. The first half of the particles with low fitness enter the next round of particle swarm reorganization directly.

(7) Genetic operator operation: According to the set parameters of each operator, the selected particles are re-selected, crossed and mutated. Through the combination of the best individual and the championship selection, the method of crossing and inserting the improved P [Z] and M [Z] is changed by the difference of inertia coefficient and acceleration constant, and the weight coefficients of each objective function are determined by the weight of each objective function in scheduling.

(8) New particle swarm optimization: Particles operated by genetic operators and those directly entering this round of operations form a new round of particle swarm optimization, which carries out the next round of cyclic search according to fitness parameter changes and particle swarm iterative operation.

(9) Judging Termination Conditions: If the Termination Conditions are satisfied, the Pareto Optimal Solution is output, otherwise step (4) is returned for the next round of search.

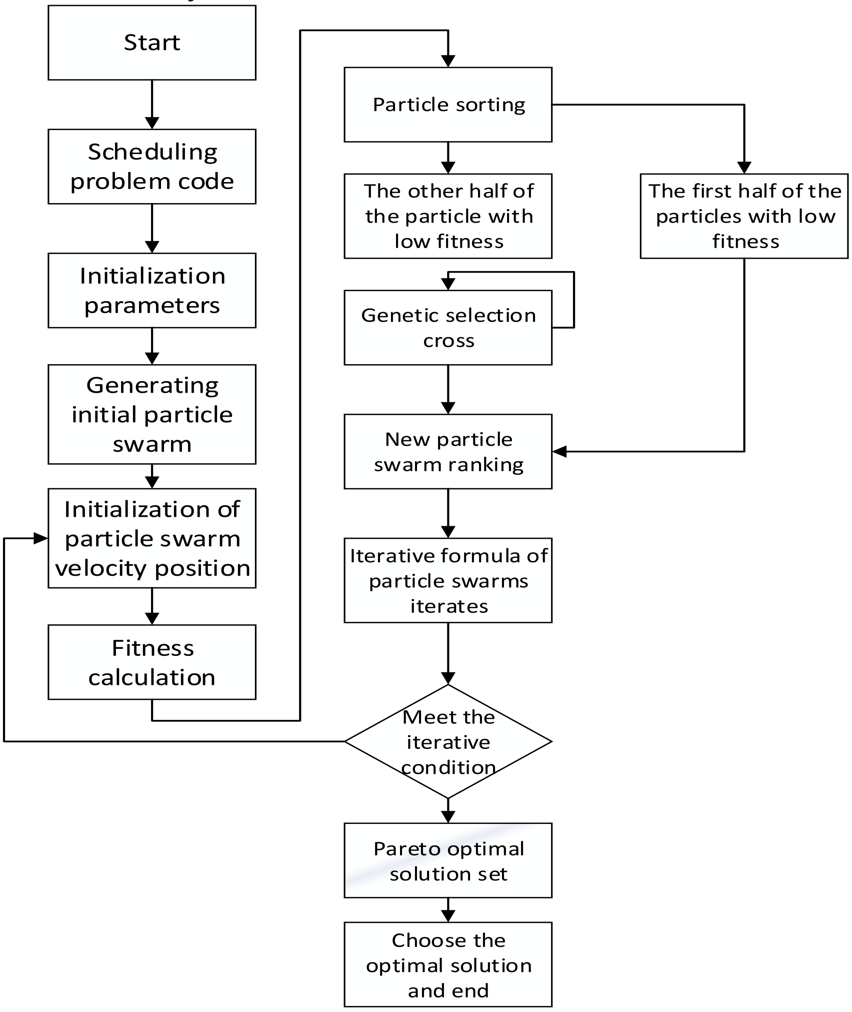

Figure 1 Mixed genetic particle population algorithm flow

4 EXAMPLE SIMULATION EXPERIMENT

4.1 Running environment

Under Intel i5, CPU's main frequency is 2.5GHz, 4G's memory and Windows XP operating system, the above algorithm is programmed with MATLAB 6.5 simulation tool.

4.2 Simulation experiment

A workshop of a bearing factory receives a batch of orders. After collecting and arranging product information, as shown in Table 4, the processing information of $6 \times 4 \times 6$ is established. The main parameters of the genetic particle swarm optimization algorithm are shown in Table 5.Because of the different performance of the machine, the cost and time of processing a certain process are also different. When using hybrid genetic particle swarm optimization, the Gantt and Pareto frontier diagrams are shown in Fig. 2 and Fig. 3. Because of the complexity of the multi-objective optimization problem, the chooser can select different groups of objective values to solve the practical problems according to the nature of Pareto solution.

Table 4 .The workpiece process corresponds to the delivery period

Workpiece number	Delivery date	Delay penalty	Punish in advance
1	15	5	2
2	14	3	1
3	13	8	3
4	16	2	1
5	20	1	1
6	19	3	2

Table 5.Main parameters of the genetic particle swarm algorithm used

GA	parameter	PSO	parameter
		Number of particle groups	50
Population size	50	scale	50
		Acceleration factor c1,c2	2
Cross rate	0.8	Inertia weight w_i	$0.2+0.2i$
Mutation rate	0.1		$i=(1,2,L\ 5)$
Genetic maximum probability	0.9		
Genetic minimum probability	0.3	Individual migration number	10
Number of iterations	50	Number of iterations	50
Total number of cycles		10	

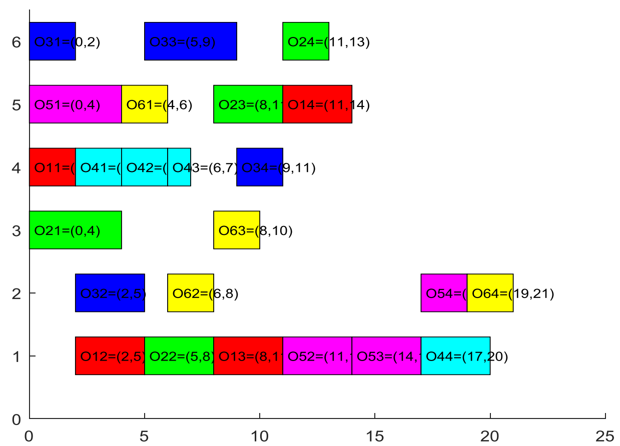

Figure2 Mixed genetic particle swarm algorithm target Gantt The abscissa is the machining time and the ordinate is the machine number.

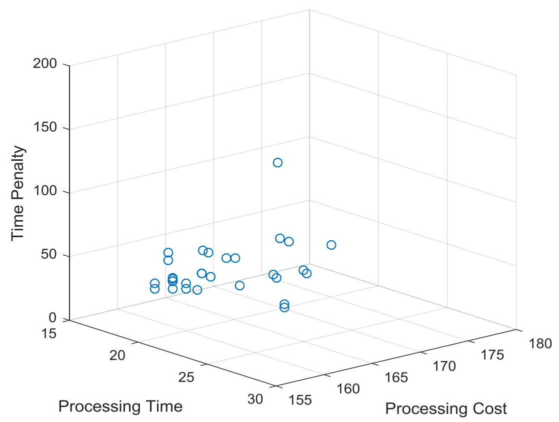

Figure 3 Pareto Front of the Objective Function of Mixed Genetic Particle Group Algorithm

5. CONCLUSION

According to the essential characteristics of scheduling, the structure-based multi-objective optimization algorithm is combined with it. The feature information of the problem is incorporated into the algorithm, and an efficient hybrid optimization algorithm framework is constructed to realize resource-constrained production and achieve the high efficiency and universality of multi-objective flexible production scheduling problem solving. The simulation results show that the algorithm has good performance compared with other algorithms.

ACKNOWLEDGMENT

the project was supported by a key research project of Higher Education of Henan Province (20A460019); and Training Plan for Young backbone Teachers of Colleges and Universities in Henan (2018GGJS194); Engineering Research Center on Intelligent Manufacturing Technology and Equipment in Henan.

REFERENCES

► Berce, Petru; Pacurar, Razvan; Balc, Nicolae. (2008). *Virtual Engineering for Rapid Product Development, International Journal of Production Research Conference: WSEAS International Conference on Engineering Mechanics, Structures, and Engineering Geology Location: Heraklion, GREECE Date: JUL 22-24, 2008, Sponsor(s): World Sci & Engn Acad & Soc,* Book Series: Mathematics and Computers in Science and Engineering, Pages: 195-+

► Coello Coello C A, Pulido G T, Lechuga M S. (2004). *Handling multiple objectives with particle swarm optimization.* IEEE Transactions on Evolutionary Computation, 8(3): 256-279.

► Demir Y, İşleyen S K. (2013). *Evaluation of mathematical models for flexible job-shop scheduling problems.* Applied Mathematical Modelling, 37(3): 977-988.

► Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. John Wiley & Sons, Vol. 7, 2-6.

► HYUN C J, KIM Y K. (1998). *A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines)*. Computers & Operations Research, (25): 7-8.

► I Kennedy J, Eberhart R C. (1995). *Particle swarm optimization.* In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ: IEEE Press, 4: 19421948

► Ishikawa S, Kubota R, Horio K. (2015). *Effective hierarchical optimization by a hierarchical multi-space competitive genetic algorithm for the flexible job-shop scheduling problem).* Expert Systems with Applications, 42(24): 9434-9440.

► KACEM I, HAMMADI S, BORNE P. (2002). *Pareto-optimality approach for flexible job scheduling problems, hybrid-ization of evolutionary algorithms and fuzzy logic)*. Mathematics and Computers in Simulation, 60(3-5): 245-276.

► Lei Deming, Yan Xinping. (2009). *Multi-objective intelligent optimization algorithm and its application*, 1-10.

► Luh P B, Hoitomt F J, Max E. (1990). Scheduling generation and reconfiguration for parallel machines. IEEE Trans Robotics and and Automation. 6(6): 687-696.

► Leordean, Dan; Radu, S. A.; Fratila, D.; et al. (2015). *Studies on design of customized orthopedic endoprostheses of titanium alloy manufactured by SLM,* International Journal of Advanced Manufacturing Technology, Vol: 79, Iss: 5-8, Pages: 905-920

► Mladenovic N, Hansen P. (1997). *Variadle neighborhood search.* Computers and Operations Research. 24:1097-1100

► Pan Feng, Chen Jie, Gan Minggang, Cai Tao, Tu Xuyan. (2006). *Model Analysis of Parcitle Swarm Optimizer,* Acta Automatica Siniva. 32(3): 368-377.

► Pang Hali. (2003). *Integrated planning and scheduling model and hybrid heuristic algorithm for flexible job shop,* Control and D ecision 18(1):34-39.

► Wang Yuping. (2011). *Theory and method of evolutionary computation [M],* Science Press, 3, 3-11.

► Wu, Xiuli. (2006). *Research on multiobiective flexible jobs scheduling problem.* Northwestern Polytechnical University. 1-10.

► Zhang Chaoyong. (2006). *A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Engineering*. Huazhong University of Science & Technology, 58-61.