A Fixture Assembly Process Planning Method Based on a Particle Swarm Algorithm

Yan Cao[1 a], Yujia Wu[1] and Sen Cao[2]

[1]School of Mechatronic Engineering, Xi'an Technological University, Xi'an, Shaanxi, China

[2]Shandong Vocational College of Economics and Business, Weifang, Shandong, China

[a] jantonyz@163.com

Keywords: Particle swarm algorithm; Optimization model; Fixture; Assembly process planning.

Abstract. In digital product assembly planning, different assembly process will direct affect the choice of assembly tools and fixtures, assembly efficiency and assembly costs. The application of a particle swarm algorithm to assembly process planning problem is studied in order to achieve the shortest completion time. Taking the shortest completion time of all the machines as the optimization goal, an optimization model is constructed. The particle swarm algorithm is realized in Matlab. An example is used to test the algorithm. The results show its effectiveness to solve assembly process planning problems.

Introduction

In digital product assembly planning, different assembly process will direct affect the choice of assembly tools and fixtures, assembly efficiency and assembly costs. For complex product assembling, you should find the best assembly solution from thousands of assembly processes. At product design stage, according to the information feedback of assembly process planning, it helps product designers improve their product design. The assembly process is an important factor in determining the complexity and reliability of an assembly process. It is of great importance to utilize the digital information of a product, generate its assembly process on a computer platform, and choose a better assembly process for an actual assembly environment.

In particle swarm algorithm [1,2], each individual will be seen as a particle of no weight and volume in n-dimensional search space, which flies at a certain speed in the search space in terms of population flying experience and its own flying experience. The particle swarm algorithm, as an efficient parallel search algorithm, preserves the global search strategy based on the population, does not depend on the feature information of the problem itself. It uses a simple evolutionary model of displacement and speed, avoids complex genetic manipulations, and adjusts only a small number of parameters. At the same time, based on its unique memory, it can dynamically track the current search situation so as to adjust its search strategies. Therefore, it has a strong global convergence capability and robustness and is very suitable for solving complex assembly process optimization problems.

Particle Swarm Optimization Model

Mechanical design optimization is the use of mathematical methods to find the optimal solution in mechanical design. Therefore, establishing a correct mathematical model in terms of an actual problem is the key to mechanical design optimization. The model describes the characteristics of the design problem and objective in mathematical form [3].

(1) Design variables

Design parameters are divided into two categories in design optimization. One is the design constants that remain unchanged in calculation process; the other is the design variables that are the parameters needed to be optimized in design process, such as ω , c_1, c_2, r_1, r_2, population size and so on in the particle swarm algorithm.

(2) Constraints

Design points constitute design space. If the value of a design point is not limited, the design space is infinite. This kind of problems is unconstrained optimization. However, in an actual mechanic design, there exist certain ranges for the design points. Constraints can be divided into inequality constraints and equality constraints. They can also be classified into boundary conditions and performance constraints according to their natural difference. Boundary conditions are mainly used to limit the ranges of the design variables and the performance constraints are mainly derived from certain natural laws, theories and formulas.

(3) Objective function

An assembly process planning problem often has many possible schemes. The task of optimization is to find out the optimal scheme. An objective function is an analytic expression represented by the design variables, and used to measure certain performance targets to be pursued. The construction and choice of the objective function is related to the usefulness of the optimization results, and will affect the degree of difficulty of optimization calculation. The objective function is measured with its size. The maximal value of the objective function f(x) is equivalent to the minimal value of the objective function -f(x). Therefore, it is represented by a unified form as below.

$$\min f(x)=\min f(x_1,x_2,...,x_n)$$

(4) Mathematical model

After determining the design variables, constraints and objective function, a standard optimization model can be expressed as below.

$$\begin{aligned} &\min f(x). \qquad (1)\\ &\text{s.t.}\quad g(x)\leq 0,\ j=1,2,\ldots,p\\ &\qquad h_k(x)=0,\ k=1,2,\ldots,q\\ &\qquad x_{imin}\leq x_i\leq x_{imax},\ i=1,2,\ldots,n \end{aligned}$$

Procedure of the Algorithm

The procedure of the particle swarm algorithm for assembly process planning is as follows.

(1) Set population size n_{pop}. Set parameter ω, c_1, c_2, r_1 and r_2. Set the maximum number of iterations N_{max}. Set the machine tools needed for each product. Determine the machine constraint matrix *M*. Set the assembly time required to assemble each product by each machine. Determine the assembly time constraint matrix *T*.

(2) Randomly generate n_{pop} initial assembly processes *X* and their initial velocities *V*. The number of the initial assembly processes and initial velocity dimensions are d that is the number of parts in the assembly process.

(3) Calculate the fitness function value of each assembly process. If the fitness function value of an assembly process is better than the fitness function value of its historical optimal assembly process P_i, the assembly process replaces the individual historical optimal assembly process P_i.

(4) Compare the fitness function values of all the individuals. Select the optimal P_i and update the global optimal assembly process P_g.

(5) For each assembly process, use the formula $V_2=\omega\cdot V_1$ [4] to calculate the effect of the speed $V_i(t)$ on the speed $V_i(t+1)$. This calculates the particle's momentum part to ensure the algorithm to possess a certain global search capability.

(6) For each assembly process, apply the formula $c_1[P_i(t)-X_i(t)]$ to calculate the effect of the individual historical best position $P_i(t)$ on the speed $P_i(t)$ at iterations t+1. That represents the part of particle's cognitive learning [5] and is the awareness and affirmation of the historical experience of the particle itself. It encourages the particles to fly towards its best position currently found.

(7) For each assembly process, apply the formula $c_2[P_g(t) - X_i(t)]$ to calculate the effect of the global optimum position P_g（t）on the speed $V_i(t+1)$ at iteration t+1. This represents the particle's social cognition[6] that guides the particles to fly towards the optimum position that the population has experienced.

(8) For each assembly process, superimpose the updated speeds of step (5), (6) and (7), $V_i(t+1) = \omega V_i(t) + c_1[P_i(t) - X_i(t)] + c_2[P_g(t) - X_i(t)]$.

(9) Apply the formula $X_i(t+1) = X_i(t) + V_i(t+1)$ to update the assembly processes.

(10) Update iteration number $t = t+1$. If $t \le N_{max}$ (N_{max} is the maximum number of iterations), go to step (3). Otherwise, go to step (11).

(11) Output the optimal assembly process P_g.

An Example and its Optimized Results

In the paper, take the following example to illustrate the application of the algorithm. The shortest completion time of all the machines is set to be the optimization goal. The optimization model is as below.

$$F(M_J) = \min（\max（T_E））. \tag{2}$$

Where, M_J is the optimal assembly process, T_E is the processing end time of each machine of each assembly process.

There are n products assembled on m machines. Herein, n=6, m=6. Time constraint matrix T represents the time needed by each working operation of each product in the assembly processes. Machine constraint matrix M represents the machine used by each working operation of each product in the assembly processes. Each row of the matrices corresponds to all the working operations needed by a product assembly process.

$$T = \begin{bmatrix} 1 & 6 & 2 & 8 & 9 & 6 \\ 8 & 5 & 7 & 9 & 7 & 4 \\ 5 & 4 & 8 & 9 & 1 & 7 \\ 6 & 7 & 5 & 3 & 4 & 5 \\ 7 & 3 & 5 & 4 & 3 & 2 \\ 3 & 7 & 2 & 6 & 4 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 5 & 1 & 2 & 4 & 3 & 6 \\ 6 & 3 & 2 & 5 & 1 & 4 \\ 2 & 4 & 3 & 1 & 6 & 5 \\ 3 & 1 & 2 & 4 & 5 & 6 \\ 4 & 2 & 5 & 6 & 1 & 3 \\ 1 & 4 & 6 & 2 & 5 & 3 \end{bmatrix}$$

According to the abovementioned algorithm and matrices, a particle swarm algorithm program is realized in Matlab to solve the example. The Gantt chart of one of the optimal solutions is shown in Fig. 1, where the abscissa represents the time required and the ordinate represents the machines. As can be seen from Fig. 1, the assembly time required for the products is 44min that is the minimum time needed. Since 0-1 evenly distributed random function formula is used, the results are not exactly the same for each run. To get reliable data, it requires multiple runs and takes the minimum.

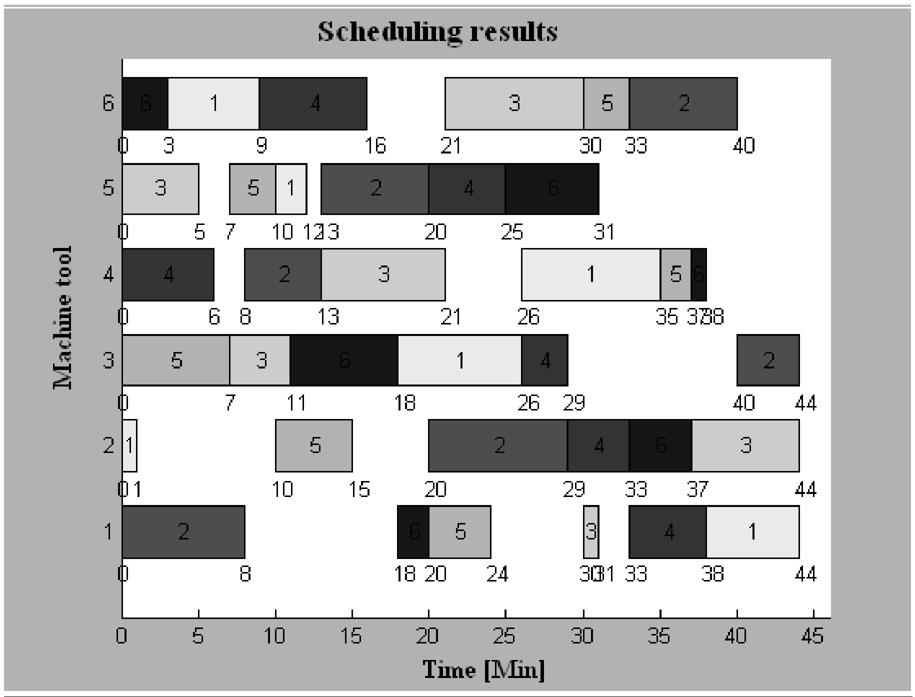

Figure 1. The Gantt chart of one of the optimal solutions

Conclusions

In the paper, the application of the particle swarm algorithm to assembly process planning is studied in order to achieve the shortest completion time. According to the time constraint matrix and machine constraint matrix, the particle swarm algorithm is programmed in the Matlab environment to determine the optimal assembly process. The algorithm is verified by an example. It is proved that the algorithm can be used to optimize the assembly process and shorten assembly process planning time.

Acknowledgments

The work is supported by Open Research Fund Program of Shaanxi Key Laboratory of Non-Traditional Machining, Key Problem Tackling Project of Shaanxi Scientific and Technological Office (2012K06-19), 2011 Key Education Innovation Project of Xi'an Technological University (11JGZ02), Natural Science Basic Research Plan in Shaanxi Province of China (2013JM7029) and Shaanxi Major Subject Construction Project.

References

[1] Dj.M. Maric, P.F. Meier and S.K. Estreicher: Mater. Sci. Forum Vol. 83-87 (1992), p. 119

[1] Y. Jin: Soft Computing Vol. 9 No. 1 (2005), p. 3

[2] S. Shan and G.G. Wang: Structural and Multidiscipinary Optimization Vol. 41 (2010), p. 219

[3] C. Lu, J.Y.H. Fuh and Y.S. Wong: Journal of Engineering Manufacture Vol. 220 No. 2 (2006), p. 255

[4] N. Iwasaki and K. Yasuda: International Journal of Innovative Computing Information and Control, Vol. 1 No. 3 (2005), p. 369

[5] R. Poli, J. Kennedy and T. Blackwell: 8th International Conference on Reliability, Maintainability and Safety (ICRMS'2009), Chengdu, China (2009), p. 1119

[6] T. Yamaguchi and K. Yasuda: Proceedings of 2006 IEEE International Conference on Systems, Man, and Cybernetics, Piscataway, NJ, USA (2006), p. 2303