

Optimal Stochastic Production Scheduling for Uncertain Re-entrant Assembly Workshops*

Nanyun Jiang^{1,2,a} and Hongsen Yan^{1,b}

¹School of Automation and the MOE Key Laboratory of Measurement and Control of Complex Systems of Engineering, Southeast University, Nanjing 210096, China

²Department of Economics and Management, Nanjing Technology University, Nanjing 210009, China

^ajiangnanyun@163.com, ^bhsyan@seu.edu.cn

Keywords: Uncertain Re-entrant; Production Scheduling; Genetic Simulated Annealing Algorithm.

Abstract. An optimal stochastic problem of production scheduling for uncertain re-entrant assembly workshops (URAWs) was studied, in which the assembly process is repeated on the basis of quality inspection result and the number of repeats is uncertain. According to analysis of URAWs, a stochastic expected value model of production scheduling was built up. In order to solve the model conveniently, the model was then converted into a deterministic model. A genetic simulated annealing algorithm based on feasible region (FR-GSAA) was designed to solve this model. Finally the result of simulation examples indicates that the model and the algorithm are feasible and practical.

Introduction

Reasonable scheduling is the core of enterprise production management. The problem of production scheduling has also been one of the hotspot issues that many scholars focus on. Previous research of scheduling concentrated on the certainty problems[1-3], while recently scheduling with uncertain factors, such as demand fluctuation[4], production capacity change[5] and process time inaccuracy[6], has gradually caught scholars' attention. In actual production, the different manufacturing processes in one job may repeat at the same equipment. The number of repeats depends on the process result of the previous operation and it is uncertain. In this paper, it is defined as uncertain re-entrance which appears primarily in manufacturing industries with high value, high precision and low output. At present, there is little research on scheduling for uncertain re-entrant assembly workshops (URAWs).

The uncertainty of re-entrance is a random event and can be generally described by stochastic variables. Thus, scheduling for URAWs is a stochastic programming problem, of which the main methods include expected value models, chance-constrained programming and dependent chance programming. The latter two methods are used to solve the problem that constraints contain stochastic variables. In this paper, the scheduling objective function contains stochastic variables. Therefore, we use the method of stochastic expected value to establish the scheduling model for URAWs, and design a genetic simulated annealing algorithm based on feasible region (FR-GSAA) to solve this model. At last, the specific numerical example proves the validity of the model and algorithm.

The Scheduling Model of URAWs

Problem Description. There are N types of products and each type needs x_n products in the assembly workshops. Every product is constructed by I_n jobs, and every job contains j operations. Each operation is assembled by corresponding assembly classes. There are M assembly classes in the assembly workshop, and every class includes G_m assembly teams that have the same function and skills. Different child jobs owning the same parent job can be assembled at the same time, while

different operations of one job must be assembled in order. After completing assembly, the product needs to be inspected strictly before left factory. If it is defective, the product must reenter into the workshop and repeat all the operations. The times of re-entrance is unlimited. The passing probability of the quality inspection is known as P_n , and the different inspection events of one product are all independent. The diagram of the assembly tree structure is shown as Fig. 1. In Fig. 1, $a-b-c$ expresses that the b th operation of job a is assembled by assembly class c .

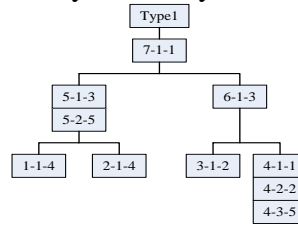


Fig. 1. Assembly tree structure of product type 1

Scheduling Model. The aim of scheduling is to minimize the total weighted completion time of all products. The assembly sequence of every operation for all products needs to be determined. Every operation needs to be assigned to the spare assembly team. And the starting time and the completion time of every assembly team also need to be arranged.

Symbols:

n : index of product types, $n = 1, 2, \dots, N$, N is the number of all the product types;

e : index of assembly quantity in product type n , $e = 1, 2, \dots, x_n$, x_n is the quantity of type n ;

i : index of job, $i = 1, 2, \dots, I_n$, I_n is the number of jobs in type n ;

l : index of re-entrant repetition, $l = 1, 2, \dots, E_n(L)$, $E_n(L)$ is the expected value of re-entrant repetitions in type n when it is qualified in quality;

j : index of operation, $j = 1, 2, \dots, J_{ni}^+$, J_{ni}^+ is the number of operations in the i th job of type n ;

m : index of assembly class, $m = 1, 2, \dots, M$, M is the number of assembly classes;

g : index of assembly team, $g = 1, 2, \dots, G_m$, G_m is the number of assembly teams in assembly class m ;

a : index of assemble times by assembly team, $a = 1, 2, \dots, A_{mg}$, A_{mg} is the maximum assemble times by team g of class m ;

O_{nelij} : the j th operation of job i in the e th product of type n when it reenters into workshop at the l th time;

$O_{(nelij)^+}$: parent operation of O_{nelij} ;

$O_{(nelij)^-}$: child operation of O_{nelij} ;

$[\bullet]^+$: $\max(0, \bullet)$.

Parameters:

β_{mg} : available time at assembly team g of class m ;

p_{nij} : assembly time of operation j in the i th job of type n ;

\tilde{t}_n : sum of quality inspection time and disassembly time in type n ;

ω_n : weight coefficients of type n ;

P_n : passing probability of quality inspection.

Decision variables:

δ_{nelijg} : 0-1 variable, if O_{nelij} is assigned to team g , then $\delta_{nelijg} = 1$, otherwise $\delta_{nelijg} = 0$;

$\rho_{nelijga}$: 0-1 variable, if O_{nelij} is assembled by team g at the a th time, then $\rho_{nelijga} = 1$, otherwise $\rho_{nelijga} = 0$;

S_{nelij} : start time of O_{nelij} ;

C_{nelij} : completion time of O_{nelij} ;

\bar{C}_{nel} : completion time of the e th product of type n when it reenters into workshop at the l th time;

\bar{S}_{mga} : start time at team g of class m at the a th time;

\bar{C}_{mga} : completion time at team g of class m at the a th time;

Model:

$$J = E \left[\text{Min} \sum_{n=1}^N \sum_{e=1}^{x_n} \omega_n \bar{C}_{n,e,E_n(L)} \right]; \quad (1)$$

$$\text{S.t. } C_{nelij} \leq S_{(nelij)^+}, \quad \forall n, e, l, i, j; \quad (2)$$

$$\bar{C}_{nel} + \tilde{t}_n \leq S_{n,e,l+1,i,1}, \quad \forall n, e, l, i; \quad (3)$$

$$C_{n,e,l,i,j-1} \leq S_{nelij}, \quad \forall n, e, l, i; j = 2, 3, \dots, J_{ni}^+; \quad (4)$$

$$S_{nelij} + p_{nij} = C_{nelij}, \quad \forall n, e, l, i, j; \quad (5)$$

$$\bar{C}_{nel} = C_{n,e,l,I_n,J_{ni}^+}, \quad \forall n, e, l; \quad (6)$$

$$\sum_{g=1}^{G_m} \delta_{nelijg} = 1, \quad \forall n, e, l, i, j; \quad (7)$$

$$\sum_{a=1}^{A_{mg}} \rho_{nelijga} = \delta_{nelijg}, \quad \forall n, e, l, i, j, g; \quad (8)$$

$$P_r(E_n(L) = K) = (1 - P_n)^{K-1} \cdot P_n, \quad K = 1, 2, \dots; \quad (9)$$

$$\delta_{nelijg} \in \{0, 1\}, \quad \forall n, e, l, i, j, g; \quad (10)$$

$$\rho_{nelijga} \in \{0, 1\}, \quad \forall n, e, l, i, j, g, a; \quad (11)$$

$$S_{nelij} \geq 0, \quad \forall n, e, l, i, j. \quad (12)$$

Eq. (1) is the objective function; Eq. (2) ensures the sequence between parent operations and child operations at the same re-entrant stage; Eq. (3) ensures the sequence between the last operation at some re-entrant stage and the first operation at the next re-entrant stage; Eq. (4) ensures the sequence between the former operation and the latter operation of one job; Eq. (5) expresses that once an operation is started, it cannot be preempted until it is completed; Eq. (6) means that completion time of the e th product in type n when it reenters into workshop at the l th time, equals to completion time of the last operation in its root node job; Eq. (7) expresses that one operation can be assigned to one assembly team only; Eq. (8) means that no two operations can be assembled simultaneously on the same assembly team; Eq. (9) is the probability distribution function of the re-entrant times. Because the different inspection events of one product are regarded as independence, $E_n(L)$ obeys the geometric distribution of parameter P_n ; Eq. (10), (11) and (12) express the range of decision variables.

Eq. (1)-(12) are the stochastic programming model. In order to solve conveniently, it is converted into a deterministic model. Since the times of re-entrance obeys the geometric distribution of parameter P_n , its expected value is as follow[8]:

$$E_n(L) = \frac{1}{P_n}. \quad (13)$$

Therefore, the model is converted into a deterministic model by plugging Eq. (13) into Eq. (1) and adding Eq. (13) as constraints. The objective function is transformed into Eq. (14), and the rest constraints are as Eq. (2)-(12):

$$C = \text{Min} \sum_{n=1}^N \sum_{e=1}^{x_n} \omega_n \mathcal{E}_{n,e,E_n(L)}. \quad (14)$$

FR-GSAA

Algorithm Flow. Genetic algorithm (GA) has strong capability of global search. However, it will be missing the optimal solution because it emphasizes the evolution between two generations. Thus its local search ability is inferior[9]. Simulation annealing algorithm (SAA) is skilled in local search, because it receives inferior solution besides optimized solution[10]. As a result, combining the advantages of GA and SAA and considering scheduling constraints, FR-GSAA is presented. Arithmetic flow of FR-GSAA is as follow:

Step 1: Initialize the parameters of FR-GSAA, such as population size, crossover rate, mutation rate, the maximum genetic generation, initial temperature T_0 , terminated temperature of simulated annealing and so on. Set the current genetic generation $gen = 1$, the times of annealing temperature updates $\theta = 1$, and the current temperature $T = T_0$;

Step 2: Initialize the population. Repair every chromosome of the population in order to satisfy all the scheduling constraints and get feasible initial population $P_\theta(gen)$;

Step 3: Select two parents from population $P_\theta(gen)$, and calculate the objective function of them as $J_{f\alpha}(gen)$ and $J_{f\beta}(gen)$. Cross two parents with crossover rate and get two new chromosomes. Calculate the objective function of the new as $J_{z\alpha}(gen)$ and $J_{z\beta}(gen)$;

Step 4: Based on the probabilities of acceptance P_α and P_β obtained by Eq. (15) and (16), determine whether to accept new chromosomes. Then get population $PZ(gen)$;

$$P_\alpha = \begin{cases} 1 & J_{f\alpha}(gen) > J_{z\alpha}(gen) \\ \exp\left[\frac{J_{z\alpha}(gen) - J_{f\alpha}(gen)}{T}\right] & J_{f\alpha}(gen) \leq J_{z\alpha}(gen) \end{cases}. \quad (15)$$

$$P_\beta = \begin{cases} 1 & J_{f\beta}(gen) > J_{z\beta}(gen) \\ \exp\left[\frac{J_{z\beta}(gen) - J_{f\beta}(gen)}{T}\right] & J_{f\beta}(gen) \leq J_{z\beta}(gen) \end{cases}. \quad (16)$$

Step 5: Select one chromosome from population $PZ(gen)$. Mutate it with mutation rate and get a new chromosome. Calculate the objective function of the new as $J_{b\gamma}'(gen)$;

Step 6: Based on the probability of acceptance P_γ obtained by Eq. (17), determine whether to accept the new chromosome. Then get new population $P_\theta(gen+1)$;

$$P_\gamma = \begin{cases} 1 & J_{b\gamma}(gen) > J_{b\gamma}'(gen) \\ \exp\left[\frac{J_{b\gamma}'(gen) - J_{b\gamma}(gen)}{T}\right] & J_{b\gamma}(gen) \leq J_{b\gamma}'(gen) \end{cases}. \quad (17)$$

Step 7: $gen = gen + 1$. If gen is greater than the maximum genetic generation, then go to Step 8, otherwise go to Step 3;

Step 8: Update current temperature T according to Eq. (18). If T is lower than terminated temperature, stop the algorithm; otherwise set $P_{\theta+1}(1) = P_\theta(gen)$, $\theta = \theta + 1$, $gen = 1$ and go to Step 3.

$$T(\theta) = \frac{T_0}{1 + \theta}. \quad (18)$$

Algorithm Design. Encoding: The chromosome contains $\sum_{n=1}^N \sum_{e=1}^{x_n} \left(\sum_{i=1}^{I_n} J_{ni}^+\right) \cdot E_n(L)$ gene blocks, each of which consists of two parts. One part represents the information of jobs. Use the string

encoding format and encode it with $n-e-l-i$. The different operations of one job use the same string, and it is expressed by its location of gene block in chromosome. The other part represents the information of assembly class and team where the operation is assembled. Use the string encoding format as before and encode it with $m.g$.

Initial Population: Construct $\sum_{n=1}^N [x_n \cdot I_n \cdot E_n(L)]$ types of gene blocks firstly and form the strings of $n-e-l-i$ in every block generate in turn within feasible region. The number of every block type is J_{ni}^+ . Then sort all the gene blocks randomly. Finally, initialize the strings of $m.g$ in every gene blocks within feasible region.

Chromosome Repairing: The initial population is generated without considering assembly constraints, therefore the chromosome must be repaired. Every gene block corresponding to the operation must be sorted by re-entrance times and arranged by the sequence between parent operation and child operation.

Selection Operation: The classical roulette method is adopted[9].

Crossover Operation: Choose randomly a gene block corresponding to non leaf node in every type of products. Then select all gene blocks corresponding to its child operations from two parent chromosomes, and exchange them.

Mutation Operation: Select a gene block randomly, and move it to a new position within the feasible range. Considering the assembly constraints, the feasible range is the intersection between the range of parent-child operation and the range of other operations of this job corresponding to this selected block.

Examples

Suppose that we have an assembly workshop with 5 assembly classes and 19 assembly teams, and the numbers of assembly teams in assembly classes are [3,5,4,3,4]. Assume that there are 5 types of products, and the assembly tree structure of type1 is shown as Fig. 1. Other parameters no longer list due to the limit of space. The requirement quantities of 5 types are [3,2,4,5,2], and the passing probabilities of the quality inspection are [0.25,0.5,0.5,0.5,0.3]. Considering characteristics of the model, we set population size to 50, generations to 100, crossover rate to 0.9, mutation rate to 0.3, initial temperature to 100, and terminated temperature to 1.

FR-GSAA proposed in this paper has been implemented through Matlab R2013b. This program is realized on an Inter Core i5 4210U CPU 2.54GHz PC with 4G memory. The optimized result of total completion time is 556.68, and the scheduling result is shown as Fig. 2. The result indicates that the model and the algorithm are feasible and effective.

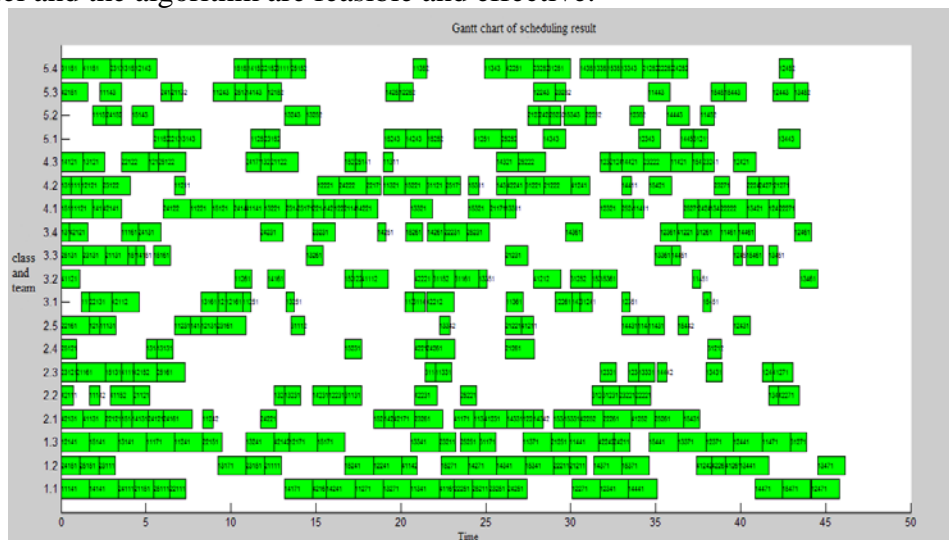


Fig. 2. Gantt chart of scheduling result

Conclusions

In this paper, we establish the scheduling model for URAWs, and propose FR-GSAA to get the optimization solution for the characteristics of model. Thereinto, we build up the stochastic expected value model aiming at the uncertainty of reentrance times, design methods of encoding, repairing, crossover operation and mutation operation based on feasible region considering the assembly constraints, and adopt genetic simulated annealing algorithm to get the solution of assembly sequence and assembly teams assignment. Finally, the result of simulation examples demonstrates the effectiveness and feasibility of the model and algorithm.

Acknowledgement

*This work is supported by a key program of the National Natural Science Foundation of China (No. 60934008), the Fundamental Research Funds for the Central Universities of China (2242014K10031), and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

- [1] J.B. Lasserre: *An Integrated Model for Job-Shop Planning and Scheduling*. Mgmt. Sci. Vol. 38 (1992), p. 1201-1211.
- [2] H.S. Yan, Q.F. Zhu, M.R. Liu, X.L. Guo and Z. M: *Integrated Production Planning and Scheduling on Automobile Assembly Lines*. IIE Trans. Vol. 35 (2003), p. 711-725.
- [3] E.D.G. Uttutia, R. Aggoune and S. Dauzere-Peres: *New Integrated Approach for Solving Multi-level Lot Sizing and Scheduling Problems*, Proceedings of 9th International Conference of Modeling, Optimization and Simulation (MOSIM'12) (Bordeaux, France 2012).
- [4] Y. Tian, H.G. Dong, X. Zou, S.S. Li and B. Wang: *Chemical Production Planning and Scheduling Integration under Demand Uncertainty*. CIESC Vol. 65 (2014), p.3552-3558.
- [5] H.P. Zhu, F.F. Hua, Y.H. Deng and H. Yin: *Extended Aggregate Production Planning Approach under Stochastic and Fuzzy Constraints*. CIMS Vol. 18 (2012), p.342-248.
- [6] L. Ding, A.M. Wang and R.X. Ning: *Job-Shop Scheduling Technology with Uncertain Processing Time*. CIMS Vol. 16 (2010), p.99-108.
- [7] B.D. Liu and R.Q. Zhao: *Stochastic Programming and Fuzzy Programming* (THU Press, Beijing 1998).
- [8] Z. Cheng, S.Q. Xie and C.Y. Pan: *Probability Theory & Mathematical Statistics* (Higher Education Press, Beijing 2010).
- [9] L. Wang: *Intelligent Optimization Algorithms with Applications* (THU Press, Beijing 2001).
- [10] Z.Q. Jiang: *Path Planning in a Static Environment Based on Genetic Simulated Annealing Algorithm* (Master's thesis, Chongqing University 2007).