

Production Scheduling and Worker Assignment Problems in Assembly Job Shop with Deterioration Effect

Tianhua JIANG

School of Transportation, Ludong University, Yantai, Shandong 264025, China

email: jth1127@163.com

Keywords: Assembly Job Shop; Production Scheduling; Worker Assignment; Deteriorating Jobs; Hybrid Genetic Algorithm

Abstract. This paper focuses on the production scheduling and worker assignment problem of an assembly job shop in static manufacturing environment. In the workshop, the processing time of each job depends on its starting time and the number of workers assigned to process it. The objective is to find the optimal scheduling and the worker assignment to minimize the production cost. A mathematical model is established by considering the characteristics of the problem. To solve it, a hybrid genetic algorithm (HGA) is proposed, where variable neighborhood search (VNS) algorithm is embedded in the genetic algorithm for scheduling improvement. Feasibility and effectiveness of the proposed model and algorithm are confirmed by the simulation results. Experimental data give a clear demonstration of the effect of two related parameters of processing time on the production cost.

Introduction

In last decades, more and more attention has been paid to the assembly job shop scheduling problem (AJSSP) which is an extension form of the classic job shop scheduling problem (JSSP). In an assembly job shop, each product has a tree structure consisting of the components and sub-assemblies that are assembled to build up the end product [1]. Both serial and parallel relationships exist between operations. As a result, in addition to the waiting time for a manufacturing resource, it is necessary to consider the time an operation may take in waiting for its mating operations before the required assembly operation can start. Due to the complexity, researches in the AJSSP are far less than those of the JSSP [2][3].

In most of the existing researches on AJSSP, processing times are all assumed to be fixed and known. However, in the real-life production, processing times of jobs may not be constant due to some activities, i.e., human or machine fatigue, learning effects, etc. In recent years, much work has been done to the scheduling problems with deteriorating jobs. When considering the deterioration effect, the actual processing time of a job is usually assumed to be a non-decreasing function of its starting time or scheduled position. Bachman, Janiak and Kovalyov [4] considered a single-machine scheduling problem with start time dependent processing times to minimize the total weighted completion time. Wu and Lee [5] studied the single-machine scheduling problem with deteriorating jobs and machine availability constraint to minimize the makespan. Wang and Xia [6] addressed the no-wait or no-idle flow shop scheduling problems with deteriorating jobs.

On the other hand, processing times are also dependent on the amount of resources (energy, money, fuel, workforce) assigned to machines. Chaudhry and Drake [7] presented a spreadsheet-based GA to solve the identical parallel machines scheduling problem where processing times are dependent on the numbers of workers assigned to machines. Shabtay and Kaspi [8] studied an extension of the classical single-machine scheduling problem of minimizing the total weighted flow time where job processing times are controllable and dependent on the resource allocation. Tseng, Liao and Huang [9] developed a NBC algorithm to solve the single machine total tardiness problem with controllable processing times.

The scheduling problems with the resource allocation or the deterioration effect have been separately studied in literatures. As far as the author knows, there exist only very few papers

considering the resource allocation and deteriorating jobs simultaneously. Wang and Wang [10] studied a single-machine earliness–tardiness scheduling problem with due date assignment, in which the processing time of a job is a function of its starting time and its resource allocation. Hsu and Yang [11] analyzed unrelated parallel-machine scheduling resource allocation problems with position-dependent deteriorating jobs.

In this paper, production scheduling and worker assignment in assembly job shop with deterioration effect are combined to be studied. A hybrid genetic algorithm (HGA) is proposed to attain a decision scheme. Feasibility and effectiveness of the proposed model and algorithm are confirmed by the simulation results. Finally, summary and further research issues are mentioned.

Problem Formulation

According to manufacture requirements, some manufacturing cells (MCs) are equipped in an assembly job shop. In each MC, machines have the same functions, and workers are assumed to have the same skill and level. The processing time of each operation depends on the numbers of workers assigned to process it and its start time. A mathematical model is established to minimize the production cost consisting of worker assignment cost and the completion cost of operations. Some assumptions are show as below.

- (1) Jobs are not allowed to be preempted.
- (2) Any two operations can not be processed simultaneously on a machine.
- (3) An operation will not be reassigned to another machine once it starts.
- (4) Due to the eligibility constraints, workers can only move in the same MC they belong to.
- (5) The number of workers on each machine will not be changed once production starts.
- (6) Setup time of machines and movement time of workers are negligible.
- (7) The normal processing time of an operation, with a deterministic maximum, is a monotone decreasing function of the number of workers, and the actual processing time is given as an increasing linear function of its start time.

$$\min Z = \sum_{m=1}^M \varphi_m |r_m - r'_m| + \sum_{i=1}^I \sum_{j=1}^{J_i} \lambda_{ij} C_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{m \in G_h} r_m \leq R_h, \forall h \quad (2)$$

$$r_{m \min} \leq r_m \leq r_{m \max}, \forall m \quad (3)$$

$$C_{ij} - S_{ij} = p_{ij}, \forall i, j \quad (4)$$

$$S_{p(ij)} - C_{ij} \geq 0, \forall i, j \quad (5)$$

$$C_{ij} - C_{i'j'} + \beta y_{iji'jm} \geq p_{ij}, \quad \forall i, i', j, j', m \quad (6)$$

$$C_{i'j'} - C_{ij} + \beta(1 - y_{iji'jm}) \geq p_{i'j'}, \quad \forall i, i', j, j', m \quad (7)$$

$$\sum_{m=1}^M x_{ijm} = 1, \forall i, j \quad (8)$$

$$p_{ijm} = [1 - \mu(1 - r_{m \min}/r_m)] \hat{p}_{ijm}, \quad \forall i, j, m \quad (9)$$

$$p_{ij} = \sum_{m=1}^M x_{ijm} p_{ijm} + \alpha S_{ij}, \forall i, j \quad (10)$$

$$x_{ijm} \in \{0, 1\}, \forall i, j, m \quad (11)$$

$$y_{iji'jm} \in \{0, 1\}, \quad \forall i, i', j, j', m \quad (12)$$

Z is the production cost; i is the index of product, $i = 1, 2, \dots, I$; j is the index of operation; J_i is the number of operations of the i th product; h is the index of MC, $h = 1, 2, \dots, H$; m is the index of machine, $m = 1, 2, \dots, M$; G_h : Machine set in the h th MC; O_{ij} is the operation j of the i th product; $O_{p(ij)}$ is the parent operation of O_{ij} ; S_{ij} is the start time of O_{ij} ; C_{ij} is the completion time of O_{ij} ; R_h is the total number of workers in the h th MC; r_m is the number of workers

assigned to the m th machine, $r_m \in [r_{m\min}, r_{m\max}]$; r'_m is the initial number of workers in the m th machine, $r'_m \in [r_{m\min}, r_{m\max}]$; λ_{ij} is the cost coefficient of the completion time of O_{ij} ; β is a large positive constant; μ is a constant ($0 \leq \mu \leq 1$) which reflects the impact of the number of workers on the normal processing times; p_{ijm} is a normal processing time of O_{ij} on the m th machine, whose maximum value is \hat{p}_{ijm} ; p_{ij} is the actual processing time of O_{ij} ; α is the common deterioration rate of operations, $0 \leq \alpha \leq 1$; x_{ijm} is a 0-1 variable, if O_{ij} is processed in the m th work group, $x_{ijm}=1$, otherwise, $x_{ijm}=0$; $y_{ij'j''m}$ is a 0-1 variable, if O_{ij} precedes $O_{i'j''}$ in the m th work group, $y_{ij'j''m}=1$, otherwise, $y_{ij'j''m}=0$.

Equation (1) indicates that the optimization objective is to minimize the production cost; constraint (2) defines that workers assigned to machines can not be more than those in the corresponding MC; constraint (3) gives the value limit of the number of workers; constraint (4) means that an operation cannot be preempted once started; constraint (5) ensures the precedence relationships between operations; constraints (6) and (7) guarantee that any two operations can not be processed simultaneously on a machine; constraint (8) means that an operation can not be reassigned to another machine once it starts; constraint (9) presents the linear relationship between the normal processing time and the number of workers; constraint (10) gives the actual processing time; constraints (11) and (12) show that x_{ijm} and $y_{ij'j''m}$ are 0-1 variables.

Hybrid Genetic algorithm

Assembly job shop scheduling problem is a special form of the job shop scheduling problem. It is almost impossible to get an optimal solution of the AJSSP directly via the mathematical model. Therefore, we propose a hybrid genetic algorithm by considering the complexity of the problem. In the HGA, the variable neighborhood search (VNS) algorithm is embedded to enforce the local search of the GA.

Figure 1 shows the BOMs of Products A and B. A-1 represents Operation 1 of Product A. Operations have three types (0,1,2) according to their locations on the BOMs. Operations on leaf node are Type 0 operations, those on root node are Type 2, and others are Type 1. In Figure 1, A-1 is a Type 2 operation, whose children are A-2, A-3 and A-4, and offspring are A-2, A-3, A-4, A-5 and A-6.

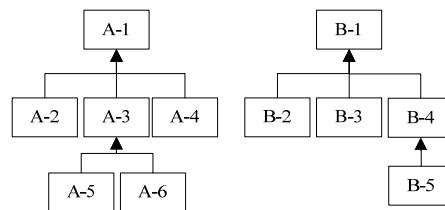


Fig.1. BOMs of Products A and B

Genetic Algorithm

1. Chromosome representation

Each chromosome is composed by three sections. The first section represents the worker assignment scheme, the second indicates the operation permutation and the third shows the machine allocation of operations. In the first section, the chromosome is divided into H blocks according to the number of MCs. The size of each block equals the number of machines in the corresponding manufacturing cell and the gene value is the number of workers. The second and third sections are of the same size, whose length is the number of operations to be scheduled and gene value are operation code and machine index respectively.

2. Initial population

For the first section, each gene is created by random selection from the value ranges of genes under the premise that constraint (2) is satisfied. The second part is generated by a random operation permutation, and the third one is created by a random selection from operations' eligible machine set.

3. Selection, Crossover and Mutation

The classic roulette is adopted to be the selection method in this algorithm. For the crossover procedure, a MC is selected randomly at first in the first section, and then the values of the two genes blocks corresponding to the selected MC are swapped between two parent chromosomes. In the second section, the order crossover is used and the two-point crossover is employed in the third.

Two different methods are adopted in the chromosome for the mutation. The single-point mutation is used in the first and third sections, and the reverse mutation approach is employed in the second section.

4. Repair

In terms of the above description, the first and the third sections are obviously always feasible, but the second one is most likely to contravene the precedence constraints between operations. Thus, repair actions should be performed to rectify infeasible chromosomes. In the repair process, if an operation has children, the operation and its offsprings should be picked out from the chromosome while keeping others' positions unchanged. The selected genes will be reordered to make the ancestor operation locate behind all its offsprings and then these selected genes will be placed back in the new sequence. The above procedure will not be terminated until the chromosome is feasible.

Variable Neighborhood Search

It is a promising approach for improving solutions by using local search in GA. To get a balance between computation speed and solution quality, the VNS is only used on the best individual of the current generation. In addition, the VNS is only applied for scheduling improvement which means that the VNS searches for better scheduling solutions under a given worker assignment. In the VNS, two neighborhood structures are designed for search.

1. Neighborhood structure N_1

In the neighborhood structure N_1 , the operation permutation in the candidate chromosome is changed, and the machine allocation remains unchanged.

Step1: Select two positions randomly in the chromosome.

Step2: Reorder the genes between the two positions by a random permutation.

Step3: Repair the sequence between the two selected positions.

Step4: Terminate the procedure.

2. Neighborhood structure N_2

In the neighborhood structure N_2 , the machine allocation is changed, in contrast, the sequence of operations remains the same.

Step1: Set $k_1 = 1$, and start the process.

Step2: Select an operation which can be processed on more than one machines at random.

Step3: Choose randomly a different machine from the original one, and assign the selected operation to it.

Step4: Set $k_1 = k_1 + 1$, if $k_1 > k_{1\max}$, go to Step 2; otherwise, terminate the procedure.

In addition, for the local search in the VNS, a threshold accepting method [12] is used based on the two neighborhood structures.

Test results

To verify the effectiveness of algorithm, we assume that there are six products (C, D, E, F, G, H) in an assemble job shop. The HGA algorithm is implemented in FORTRAN and executed on a PC with Intel Core i5-2400 CPU @ 3.10GHz and 3GB main memory under WINXP. Parameters of the algorithm are given in Table 1.

Table 1. Parameters of HGA

Population size	Generation	Crossover rate	Mutation rate	k_1
50	300	0.8	0.2	10

The BOM of Product C is illustrated in Figure 2. The resource allocation of manufacturing cells

is presented in Table 2. The initial worker assignment of machine is [3,5,5,3,4,4,4,3,3,3,4,2,4,3,4,3,3,3,3]. The maximum processing time of operations are selected randomly in [2,25].

Table 2. Resource allocation of MCs

Manufacturing cell (MC)	Number of machines	Machine index	Total number of workers in MC	Value range of the number of workers assigned to machine
MC1	2	1,2	8	[2,6]
MC2	2	3,4	10	[2,8]
MC3	1	5	6	[3,6]
MC4	3	6,7,8	10	[2,5]
MC5	2	9,10	8	[2,6]
MC6	3	11,12,13	12	[1,4]

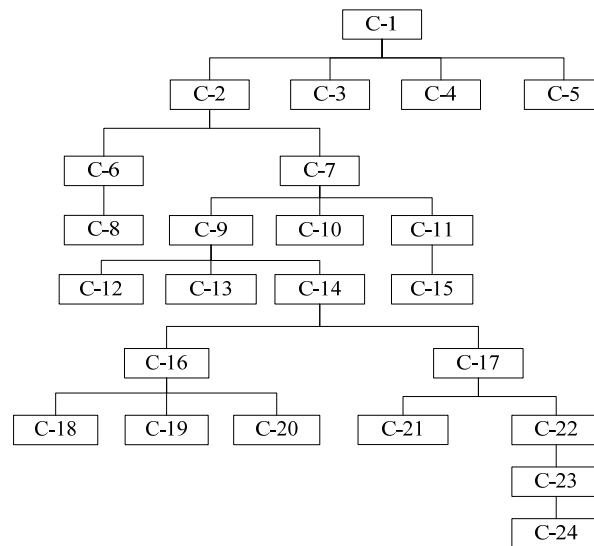


Fig. 2. BOM of Product C

To verify the feasibility of the proposed algorithm and analysis the effect of two key parameters on the production cost, extensive experiments are preformed with different values of α and μ . The experimental data are shown in Table 3. It indicates that: 1) for a given value of α , the workshop with a larger value of μ promises a lower production cost due to the effect of worker adjustment on the normal processing time reduction; 2) for a fixed value of μ , as α increases, the production cost also increases due to the deteriorating processing times.

Table 3. Effect of α and μ on production cost

μ	α	1	2	3	4	5	6	7	8	9	10	Average
0.	0.	69630	71577	68641	69052	70294	70115	69018	69440	71388	68893	69805
	1											
	2	13198	13841	13463	13167	13873	13155	13696	12994	13583	13545	134521
2	0.	8	9	9	7	9	0	3	7	7	5	
	1											
	2	35789	33284	33883	33969	34105	35016	34539	33407	34853	33001	341851
3	0.	3	5	8	8	3	6	0	9	7	4	
	1											
	2	60398	60425	61248	60206	59511	58255	59414	61302	61096	60601	60246
0.	0.	11516	11522	11953	11559	12094	11017	11613	10976	11546	11702	115504
	1											
	2	5	8	3	7	6	9	5	5	5	8	
5	0.	26772	28061	27422	28100	26349	27661	27692	26344	28198	28931	275534
	1											
	2	1	6	7	4	3	6	3	6	7	1	
0.	0.	51847	51458	50010	52913	49752	50557	48682	51955	52136	51049	51036
	1											
	2	10085	10620	10062	10593	10567	10126	99520	99469	10371	10119	102445
8	0.	0	5	1	7	2	6			9	4	
	1											
	2	20823	21553	19936	20958	20071	20942	21042	20722	21938	21519	209510
3	0.	7	9	9	5	2	4	6	8	5	0	
	1											
	2											

Conclusion

This paper considered the production scheduling and worker assignment problem in an assembly job shop with deteriorating jobs. A mathematical model is established and a hybrid genetic algorithm (HGA) is developed. Simulation results demonstrate that the proposed model and algorithm are effective and feasible. In addition, extensive experiments are preformed to analyze the effect of related parameters on the production cost. By considering the real-life production environment, movement times of workers and setup times between operations should be incorporated in the future study.

Acknowledgement

In this paper, the research was sponsored by the Talent Introduction Project of Ludong University (Project No. 32660301).

References

- [1] Pathumnakul S, Egbelu PJ. An algorithm for minimizing weighted earliness penalty in assembly job shops[J]. *International Journal of Production Economics*, 2006 103 (1) 230-245.
- [2] Omkumar M, Shahabudeen P. Ant colony optimization for multi-level assembly job shop scheduling[J]. *International Journal of Manufacturing Research*, 2009 4 (4) 410-427.
- [3] Pereira M, Santoro MC. An integrative heuristic method for detailed operations scheduling in assembly job shop systems[J]. *International Journal of Production Research*, 2011 49 (20) 6089-6105.
- [4] Bachman A and so on. Minimizing the total weighted completion time of deteriorating jobs. *Information Processing Letters*, 2002 81 (2) 81-84.
- [5] Wu CC, Lee WC. Scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine[J]. *Information Processing Letters*, 2003 87 (2) 89-93.
- [6] Wang JB, Xia ZQ. Flow shop scheduling with deteriorating jobs under dominating machines[J]. *Omega*, 2006 34 (4) 327-336.
- [7] Chaudhry IA, Drake PR. Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms[J]. *The International Journal of Advanced Manufacturing Technology*, 2009 42 (5-6) 581-594.
- [8] Shabtay D, Kaspi M. Minimizing the total weighted flow time in a single machine with controllable processing times[J]. *Computers & Operations Research*, 2004 31 (13) 2279-2289.
- [9] Tseng CT and so on. Minimizing total tardiness on a single machine with controllable processing times[J]. *Computers & Operations Research*, 2009 36 (6) 1852-1858.
- [10] Wang XY, Wang JJ. Single-machine due date assignment problem with deteriorating jobs and resource-dependent processing times[J]. *The International Journal of Advanced Manufacturing Technology*, 2013 67 (1-4) 255-260.
- [11] Hsu CJ, Yang DL. Unrelated parallel-machine scheduling with position-dependent deteriorating jobs and resource-dependent processing time[J]. *Optimization Letters*, 2014 8 (2) 519-531.
- [12] Bouffard V, Ferland JA. Improving simulated annealing with variable neighborhood search to solve the resource-constrained scheduling problem[J]. *Journal of Scheduling*, 2007 10 (6) 375-386.