Due-Date Based Multi-period Part Selection for Flexible Manufacturing Systems with Controllable Processing Times

Issa B. Zeid, Hyoung-Ho Doh, Jeong-Hoon Shin, and Dong-Ho Lee[(⊠)]

Department of Industrial Engineering, Hanyang University, Seoul, Korea
leman@hanyang.ac.kr

Abstract. This study addresses a multi-period part selection problem for flexible manufacturing systems in which processing times are controllable. The problem is to determine the set of parts and their processing times while satisfying the processing time and the tool magazine capacities in each period of a planning horizon. The objective is to minimize the sum of processing, earliness/tardiness, subcontracting and tool costs. Practical considerations such as available tool copies and tool lives are also considered. An integer programming model is developed, and two-phase heuristics are proposed in which an initial solution is obtained by a greedy heuristic under initial processing times and then it is improved using local search methods while adjusting processing times. Computational experiments were done on a number of test instances, and the results are reported.

Keywords: Flexible manufacturing systems · Controllable processing times
Part selection · Heuristics

1 Introduction

Flexible manufacturing system (FMS) is an automated manufacturing system that consists of numerical control machines and an automated material handling/storage system, which are controlled by a computer control system. Each machine has an automatic tool changer that can interchange cutting tools automatically, which allows consecutive operations to be performed with negligible setup times. Therefore, an FMS is capable of processing various part types simultaneously with higher utilization.

Part selection, alternatively called batching in the literature, is the problem of selecting the parts to be produced during the upcoming planning period. Most previous studies on FMS part selection propose single-period models that determine a set of parts to be produced simultaneously during an upcoming period. See Hwang and Shogan [1], Kim and Yano [2] and Bilge et al. [3] for examples. To obtain better solutions over the planning horizon with multiple periods, some articles extend the single-period models to multi-period ones. See Stecke and Toczylowski [4] and Lee and Kim [5] for examples.

This study focuses on a multi-period part selection problem with controllable processing times, which is the problem of determining a set of parts and their processing times in each period of a planning horizon. The controllable processing times imply that part processing times are not given, but can be changed to cope with system requirements such as energy consumption, scheduling performances, and so on. See Nowicki and Zdrzalka [6] for the detail of the controllable processing time concept.

To represent the problem mathematically, an integer programming model is developed. Then, due to the complexity of the problem, two-phase heuristics are proposed in which an initial solution is obtained by a greedy heuristic under initial processing times and then it is improved using local search methods while adjusting part processing times. Computational experiments were done, and the results are reported.

2   Problem Description

The FMS considered in this study consists of one or more numerical control machines, each of which has a tool magazine of a limited tool slot capacity. The machines can process different parts with negligible setup times if tooled differently in the tool magazines. To produce a part, several tools are required and each tool requires one or more slots in the tool magazine, where each tool has multiple copies with a limited life. Also, processing times of parts are controllable with different processing costs.

The problem is to determine the set of parts to be produced in each period of a planning horizon and their processing times while satisfying the constraints on processing time capacity, tool magazine capacity, tool copies and tool lives. The objective is to minimize the sum of part processing, earliness/tardiness, subcontracting and tool costs. The problem can be represented as the following integer programming model. The notations used are summarized below.

Parameters

d_i due-date of part i

cp_i^j processing cost of part i under the jth available processing time

cd_{ih} earliness or tardiness cost of part i incurred when it is assigned to period h, i.e.

$$cd_{ih} = \begin{cases} \epsilon_i \cdot (d_i - h), & \text{if } h \leq d_i \\ \tau_i \cdot (h - d_i), & \text{if } h \geq d_i, \end{cases}$$

where ϵ_i and τ_i are per-period earliness and tardiness cost of part i, respectively

cs_i subcontracting cost of part i

ct_t cost of tool type t

p_i^j jth available processing time of part i

$$\left(p_i^1 \leq p_i^2 \leq \cdots \leq p_i^{J_i} \text{ and } cp_i^1 \geq cp_i^2 \geq \cdots \geq cp_i^{J_i} \text{ for all } i. \right)$$

TC_t number of available copies of tool type t

TL_t life of tool type t

s_t	number of slots required by tool type t
Φ_t	set of parts that require tool type t
L	aggregated processing time capacity in period h
S	aggregated tool magazine capacity in period h

Decision variables

x_{ih}^j = 1 if part i is assigned to period h with the jth available processing time, and 0 otherwise

y_{th} number of copies for tool type t used in period h

Now, the integer programming model is given below. The detailed explanation is skipped here due to the space limitation. The problem [P] is NP-hard because the problem with fixed part processing times can be reduced to the generalized assignment problem that is known to be NP-hard [7].

$$[\text{P}]\text{Minimize} \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{h=1}^{H} cp_i^j \cdot x_{ih}^j + \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{h=1}^{H} cd_{ih} \cdot x_{ih}^j$$
$$+ \sum_{i=1}^{I} cs_i \cdot \left(1 - \sum_{j=1}^{J_i} \sum_{h=1}^{H} x_{ih}^j\right) + \sum_{t=1}^{T} \sum_{h=1}^{H} ct_{th} \cdot y_{th}$$

subject to

$$\sum_{j=1}^{J_i} \sum_{h=1}^{H} x_{ih}^j \leq 1 \qquad \text{for all } i \qquad (1)$$

$$\sum_{i=1}^{I} \sum_{j=1}^{J_i} p_i^j \cdot x_{ih}^j \leq L \qquad \text{for all } h \qquad (2)$$

$$\sum_{t=1}^{T} s_t \cdot y_{th} \leq S \qquad \text{for all } h \qquad (3)$$

$$\sum_{h=1}^{H} y_{th} \leq TC_t \qquad \text{for all } t \qquad (4)$$

$$\sum_{i \in \Phi_t} \sum_{j=1}^{J_i} p_i^j \cdot x_{ih}^j \leq TL_t \cdot y_{th} \qquad \text{for all } t \text{ and } h \qquad (5)$$

$$x_{ih}^j \in \{0, 1\} \qquad \text{for all } i, j \text{ and } h \qquad (6)$$

$$y_{th} \geq 0 \text{ and integer} \qquad \text{for all } t \text{ and } h \qquad (7)$$

3 Solution Approach

3.1 Phase I: Obtaining an Initial Solution

An initial solution is obtained by sorting the parts in the non-increasing order of subcontracting costs and allocating each part in the sorted list to the period with the smallest earliness/tardiness cost while satisfying the constraints.

3.2   Phase II: Improvement

Before explaining the improvement methods, the processing time adjustment method when a part cannot be moved to a period due to the time capacity is explained.

Adjusting Part Processing Times. If a part cannot be allocated to a period due to the processing time capacity, it is checked if the part can be allocated to the period after adjusting the processing times of the part as well as the parts allocated already to the period. Specifically, it is checked if a part to be moved can be allocated to a period while improving the current solution after its processing time is reduced. If it is not possible, the processing times of the parts allocated already to the period are changed one-by-one and check the possibility of moving the part to the period. For this purpose, the following rules to select the part to be moved are tested. In the following, $j(i)$ denotes the index for the processing time selected for part i.

CTR (cost/time ratio): select part i^* such that

$$i^* = \mathrm{argmin}_{i \in X_h} \left\{ \left(cp_i^{j(i)-1} - cp_i^{j(i)} \right) \Big/ \left(p_i^{j(i)} - p_i^{j(i)-1} \right) \right\}$$

MCI   (minimum   cost   increase):   select   part i^* such   that
$i^* = \mathrm{argmin}_{i \in X_h} \left\{ cp_i^{j(i)} - cp_i^{j(i)-1} \right\}$

MTD   (maximum   time   decrease):   select   part i^* such   that
$i^* = \mathrm{argmax}_{i \in X_h} \left\{ p_i^{j(i)} - p_i^{j(i)-1} \right\}$

Improvement. The improvement method consists of interchange, insertion, perturbation and reallocations of subcontracted parts in sequence. For the current solution, let P_T and P_E denote the set of tardy and early parts, respectively.

Interchange Method. The parts in P_T (P_E) are sorted in the non-increasing order of the tardiness (earliness) costs. Then, according to the sorted list, each part is interchanged with the ones in the periods with less earliness (tardiness) costs than that of the part considered while adjusting the processing times of the parts to be interchanged and included in the periods that the parts are to be inserted and the best one is selected.

Insertion Method. Each part in P_T and P_E is removed from its original period and then it is inserted to another feasible period that reduces the total cost. To reduce the search space, we consider the periods with less earliness or tardiness costs than that of the part to be moved for the parts in P_T and P_E. The following two methods are tested.

BI (best improvement): The parts in P_T (P_E) are sorted in the non-increasing order of their tardiness (earliness) costs. Then, from the first part in the sorted list, it is removed from its original period and then allocated to the first feasible period that improves the current solution while adjusting the part processing times.
HI (hybrid improvement): From the first part, the first and the best periods that improve the current solution are found and then the better one is selected.

Perturbation Method. Each of the parts allocated to its due-date period is moved to the period with the minimum increase in cost and then the part with the largest earliness or tardiness cost is moved from the original period to the due-date period.

Reallocation Method. The subcontracted parts are sorted in the non-increasing order of subcontracting costs. Then, from the first to the last part in the sorted list, it is inserted to the first feasible period that improves the current solution while adjusting the processing times of relevant parts, where the insertions are done from the due-date to other periods in the non-decreasing order of earliness and tardiness cost.

4   Computational Results

Computational experiments were done to identify the best one among the 6 combinations of 3 processing time adjustment methods (CTR, MCI and MTD) and 2 variations of the improvement methods (BI and HI). The algorithms were coded in C++ and the tests were done on a PC with Intel Core i7 CPU at 3.40 GHz.

The first test was done for small-sized test instances and reports the percentage gaps from the optimal solution values, i.e. $100 \cdot \left(C_a - C_{opt}\right)/C_{opt}$, where C_a is the objective value obtained from combination a and C_{opt} is the optimal values obtained from the CPLEX with a time limit of 3600 s. For this test, 60 instances with 5 periods were generated randomly, i.e. 10 instances for each of the 6 combinations of 3 levels for the number of parts (20, 30 and 50) and 2 levels of the number of tools (tight and loose). The detailed data generation method is skipped due to the space limitation.

Test results are summarized in Table 1 that shows the number of instances that the CPLEX gave optimal solutions within 3600 s and the average percentage gaps. Although no one dominates the others, CTR-HI works better than the others and its overall average gap was 1.80. Finally, the CPU seconds of the two-phase heuristics are not reported since all the test instances were solved within 3 s.

Table 1.   Test results for small-sized test instances

Number of parts	Number of periods	Tooling tightness	N_{opt}^{a}/ CPU_{opt}^{b}	CTR[c]		MCI[c]		MTD[c]	
				BI	HI	BI	HI	BI	HI
20	5	Tight	10/28.9	3.70/0.6[d]	3.70/0.8	2.56/0.6	2.56/0.7	2.84/0.7	2.84/0.8
		Loose	10/602.2	3.80/1.5	3.53/3.3	4.37/0.9	4.49/1.0	4.00/1.0	4.35/1.1
30	5	Tight	10/81.6	0.87/1.7	0.83/2.3	1.26/2.2	1.26/2.6	1.37/2.1	1.37/2.4
		Loose	8/162.1	1.06/1.8	1.14/2.4	1.45/2.3	1.45/2.6	1.04/2.3	1.26/2.5
50	5	Tight	9/963.7	1.06/2.9	0.84/3.4	0.78/3.0	0.78/3.2	1.10/2.8	1.11/2.5
		Loose	8/1206.5	0.65/4.2	0.75/5.0	0.75/3.7	0.69/3.6	0.59/3.4	0.59/3.8
Average				1.86	1.80	1.86	1.87	1.82	1.92

[a]Number of instances that the CPLEX gave the optimal solutions (out of 10 instances)
[b]Average CPU seconds of the CPLEX (for the instances that the optimal solutions were obtained)
[c]Processing time adjustment methods: CTR (cost/time ratio), MCI (minimum cost increase), MTD (maximum time decrease)
[d]Average percentage gap/CPU second out of 10 instances

The second test was done on large-sized instances. Because the optimal solutions could not be obtained, we compared them using the relative performance ratio, i.e.

$$100 \cdot (C_a - C_{best})/C_{best},$$

where C_{best} is the best objective value among those obtained by all combinations. Table 2 shows the test results that are similar to those for the small-sized instances.

Table 2. Test results for large-sized test instances

Number of parts	Number of periods	Tightness of tooling	CTR		MCI		MTD	
			BI	HI	BI	HI	BI	HI
30	10	Tight	2.49/2.2[a]	2.44/2.7	1.94/2.2	1.18/2.5	3.19/2.3	3.19/2.6
		Loose	0.85/2.6	0.85/2.9	2.96/2.4	2.54/2.8	3.07/2.7	3.18/2.8
	15	Tight	2.85/2.5	2.85/2.8	2.98/2.6	3.18/3.1	4.55/2.7	4.55/3.1
		Loose	1.92/2.8	1.92/3.1	3.76/2.8	3.76/3.6	0.48/2.8	0.48/3.3
50	10	Tight	1.33/4.9	1.33/5.4	2.85/4.7	2.85/5.5	0.82/5.3	0.85/5.8
		Loose	1.18/5.0	1.10/5.6	0.78/5.6	0.78/6.0	1.58/5.1	1.49/5.6
	15	Tight	1.43/5.8	1.68/6.3	2.14/5.9	2.14/7.0	1.26/5.9	1.90/6.8
		Loose	1.81/7.4	1.77/8.4	1.80/6.7	1.80/7.8	2.56/7.1	2.22/8.1
100	10	Tight	0.55/21.2	0.38/24.0	1.13/18.9	1.14/24.9	0.63/15.1	0.44/26.8
		Loose	0.51/18.5	0.50/18.9	0.56/18.6	0.69/21.7	0.30/16.0	0.30/15.3
	15	Tight	0.57/19.1	0.57/20.8	1.78/15.4	1.70/17.9	0.74/15.1	0.74/19.9
		Loose	1.03/14.4	1.03/16.3	0.62/15.3	0.74/17.9	0.86/13.1	0.86/16.9
Average			1.49	1.46	2.22	2.15	1.89	1.90

[a]Average relative performance ratio/CPU second out of 10 instances

5 Concluding Remarks

We considered multi-period part selection for FMSs with controllable processing times. The problem is to determine the set of parts and their processing times that satisfy the processing time and the tool magazine capacities in each period of a planning horizon for the objective of minimizing the sum of part processing, earliness/tardiness, sub-contracting and tool costs. The number of available tool copies, tool life restrictions and tool sharing were also considered. An integer programming model was developed, and then two-phase heuristics were proposed in which an initial solution is obtained by a greedy algorithm under initial processing times and then it is improved using two local search methods. Computational experiments were done on a number of test instances, and the best ones were reported.

For further research, it is needed to develop meta-heuristics that can improve the solution quality especially for large-sized instances. Also, the problem can be integrated with other problems such as loading and scheduling.

4, Title: Development of a jig-center class 5-axis horizontal machining system capable of over 24 h continuous operation).

References

1. Hwang, S., Shogan, A.W.: Modelling and solving an FMS part-type selection problem. Int. J. Prod. Res. **27**(8), 1349–1366 (1989)
2. Kim, Y.-D., Yano, C.A.: A due date-based approach to part type selection in flexible manufacturing systems. Int. J. Prod. Res. **32**(5), 1027–1043 (1994)
3. Bilge, Ü., Albey, E., Beşikci, U., Erbaur, K.: Mathematical models for FMS loading and part type selection with flexible process plans. Eur. J. Ind. Eng. **9**(2), 171–194 (2015)
4. Stecke, K.E., Toczylowski, E.: Profit-based FMS dynamic part type selection over time for mid-term production planning. Eur. J. Oper. Res. **63**(1), 54–65 (1992)
5. Lee, D.-H., Kim, Y.-D.: A multi-period order selection problem in flexible manufacturing systems. J. Oper. Res. Soc. **49**(3), 278–286 (1998)
6. Nowicki, E., Zdrzalka, S.: A survey of results for sequencing problems with controllable processing times. Discret. Appl. Math. **26**(2–3), 271–287 (1990)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, New York (1979)