Loading Algorithms for Flexible Manufacturing Systems with Partially Grouped Unrelated Machines and Tooling Constraints

H.-W. Kim, J.-S. Kim, J.-M. Yu, H.-H. Doh
D.-H. Lee†
Department of Industrial Engineering
Hanyang University
Seoul, Korea
e-mail: leman@hanyang.ac.kr

S.-H. Nam
Manufacturing Convergence R&D Department
Korea Institute of Industrial Technology
Ansan, Korea

***Abstract*—This paper considers the loading problem for flexible manufacturing systems with partially grouped machines, i.e., machines are tooled differently, but multiple machines can be assigned to each operation. Loading is the problem of allocating operations and their associated cutting tools to machines for a given set of parts. As an extension of the existing studies, we consider unrelated machines, i.e., processing time of an operation depends on the speed of the machine where it is allocated. Also, we consider the practical constraints associated with cutting tools: (a) tool life restrictions; and (b) available number of tool copies. An integer linear programming model is suggested for the objective of balancing the workloads assigned to machines. Then, due to the complexity of the problem, we suggest two-stage heuristics in which an initial solution is obtained and then it is improved. The heuristics were tested on some test instances, and the results are reported.**

Keywords-flexible manufacturing system; production planning; loading; unrelated machine; tooling; heuristics

I. INTRODUCTION

A flexible manufacturing system (FMS), which pursues the productivity and the flexibility at the same time, can be defined as an automated manufacturing system consisting of computer numerically controlled machines and auxiliary equipment such as inspection and washing stations that are interconnected by means of an automated material handling and storage system, all controlled by a computer system. Significant improvement was reported from FMSs with respect to various measures such as quality, costs, inventory levels, space requirements, lead times, scrap rates, etc.

Among the FMS operation problems, we focus on loading, one of system setup problems. In general, the FMS loading problem is concerned with allocating operations and their cutting tools to machines for given set of parts. Note that the loading problem is closely related with other decision problems such as part type selection (batching), scheduling, and control.

From a pioneering research of Stecke [1], a number of researchers considered various loading problems. According to Lee and Kim [2], they can be classified according to the ways to grouping machines: no, total, and partial groupings. Most previous research articles consider total grouping in which machines are partitioned into groups and all machines in the same group are identically tooled. See Stecke [1], Kim and Yano [3], Swarnkar and Tiwari [4], Kumar *et al*. [5], and Biswas and Mahapatra [6] for the loading problems on no or total machine grouping configuration. Unlike these, Stecke and Raman [7] considered the loading problem in partial machine grouping, in which machines are tooled differently, but multiple machines can be assigned to each operation. Later, Lee and Kim [2] considered the problem under the partial machine grouping configuration and suggested various heuristics that can give near optimal solutions. In this paper, they showed that partial grouping gives generally better performances since it helps to cope with system disturbances such as machine breakdowns more easily by providing more routing flexibility. See Grieco *et al*. [8] for a literature review on the FMS loading problems.

This paper considers the loading problem under partial machine grouping since it performs better than the other configurations. In other words, the problem is to allocate operations and their associated cutting tools to machines in the case that machines are tooled differently, but multiple machines can be assigned to each operation. In fact, we extend the existing model of Lee and Kim [2] since it is too simple to consider practical constraints associated with machines and tools. They are: (a) unrelated machines, i.e., processing time of an operation depends on the machine where it is allocated; (b) tool life restrictions; and (c) available tool copies. Although some recent articles consider those practical constraints, they don't consider them at the same time. For example, unrelated and dedicated machines are considered by Swarnkar and Tiwari [4] and Biswas and Mahapatra [6] and the tooling constraints by Kumar *et al*. [5]. Instead, we consider the practical constraints at the same time, especially under the partial grouping configuration. Note that this study was motivated from

† Corresponding author

978-1-4244-5567-6/10/$26.00 ©2010 IEEE

3CA 2010

a research project on a real FMS.

To represent the problem mathematically, an integer linear programming model is suggested for the problem with the objective of balancing the workloads among machines, i.e., minimizing the maximum workload. Then, due to the complexity of the problem, we suggest two-stage heuristics in which an initial solution is obtained and then it is improved. The heuristics were tested on various randomly generated test instances, and the results are reported.

II. Problem Description

The FMS considered here has several unrelated machines, each with an automatic tool changer and a tool magazine of a limited capacity. These machines can perform different sets of operations if tooled differently. However, the processing time of an operation may be different according to the speed of the machine to which the operation is allocated. To perform an operation, one or more tools are required, and each tool requires one or more slots in the tool magazine. Two or more operations may share the same tools (a tool may be used for two or more operations) in the system, called tool sharing or tool commonality in the literature. Also, the FMS has a limited number of tools for each tool type, i.e., tool copy limitations, and each tool has different lifespan, i.e., tool life restrictions.

The loading problem considered here is to allocate operations and their required tools to the unrelated machines for the objective of minimizing the maximum workload of the machines, i.e., balancing the workloads assigned to machines. Note that balancing the workloads is closely related with maximizing the system throughput. The main constraints of the problem are tool magazine capacity, number of available tool copies, and tool life restrictions. It is assumed that a set of part types has been selected to be produced at the same time during the upcoming production period, and production quantities for the part types have been determined.

For a clear description of the problem, we formulate the loading problem as an integer linear programming model. Note that the model is an extension that of Lee and Kim [2] by incorporating the additional considerations. The following notations are used in the formulation.

Parameters

i	index for part types, $i = 1, 2,\dots I$
j	index for operations, $j = 1, 2,\dots J_i$
m	index for machines, $m = 1, 2,\dots M$
t	index for tools, $t = 1, 2,\dots T$
D_{ij}	processing requirement (in units) of operation j for part type i
p_{ijm}	processing time of operation j for part type i on machine m
TS_t	number of tool slots needed for tool t
TL_t	tool life for tool t (unit in time)
TC_t	number of available tool copies for tool t
TMC_m	tool magazine capacity on machine m
$\Omega(m^*)$	set of essential part types that must be processed on dedicated machine m^*
a_{ijt}	= 1 if operation i for part type j requires tool t, and 0 otherwise

Decision variables

w_{ijm}	processing requirement (in units) of operation j for part type i assigned to machine m
z_{tm}	number of tool t assigned to machine m

Now, the integer linear program is given below.

[P] Minimize Z

subject to

$$\sum_{i=1}^{I}\sum_{j=1}^{J_i} p_{ijm}\cdot w_{ijm} \le Z \qquad \text{for all } m \qquad (1)$$

$$\sum_{m=1}^{M} w_{ijm} = D_{ij} \qquad \text{for all } i \text{ and } j \qquad (2)$$

$$\sum_{t=1}^{T} TS_t\cdot z_{tm} \le TMC_m \qquad \text{for all } m \qquad (3)$$

$$\begin{cases} \text{If } m=m^*, & \sum_{i\in\Omega(m^*)} w_{ijm} = \sum_{i\in\Omega(m^*)} D_{ij} \\ \text{Otherwise}, & \sum_{i\in\Omega(m^*)} w_{ijm} = 0 \end{cases}$$

$$\text{for all } j,\ m \text{ and } \Omega(m^*)\neq\phi \qquad (4)$$

$$\sum_{m=1}^{M} z_{tm} \le TC_t \qquad \text{for all } t \qquad (5)$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J_i} a_{ijt}\cdot p_{ijm}\cdot w_{ijm} \le TL_t\cdot z_{tm}$$

$$\text{for all } t \text{ and } m \qquad (6)$$

$$w_{ijm}, z_{tm}\in Z^{+} \qquad \text{for all } i,\ j,\ t \text{ and } m \qquad (7)$$

In the above formulation, the objective together with constraint (1) minimizes the maximum workload, i.e., balancing the workloads. Processing requirements of operations can be satisfied by constraint (2), and tool magazine capacity constraints are represented by (3). Constraint (4) represents essential part types and their dedicated machines, i.e., certain part types must be processed on specific machines. Constraints (5) and (6) represent the number of available tool copies for each tool type and tool life restrictions, respectively. Finally, constraint (7) represents the conditions of the decision variables.

III. Solution Algorithms

Two heuristics are suggested in this study. As stated earlier, the heuristics consists of two stages: (a) obtaining an initial solution; and (b) improvement.

A. LPT based heuristic

In this heuristic, the initial solution is obtained as follows. First, the number of machines that will be assigned to each operation is set to be equal and the processing requirement of an operation is divided into the same number of batches for all operations. In other words, the number of batches is checked for all alternatives for the number, i.e. from one to M (total number of machines in the system). Then, these batches are allocated to machines by the LPT (longest processing time) algorithm, a single-pass algorithm for the bin-packing problem, while considering the tool copy and tool life constraints.

The procedure to obtain the initial solution can be summarized below.

Procedure 1. (Obtaining an initial solution: LPT)

Step 1. Let $Z = \infty$ and $m = 1$. (Z is the solution value, i.e., the maximum workload and m is the number of machines to be assigned to each operation.)

Step 2. If $m > M$ (number of machines), stop. Otherwise, go to Step 3.

Step 3. Constitute batches for each operation by dividing processing requirement (in units) of an operation by m (to a nearest integer).

Step 4. Repeat the following steps until all batches are allocated to the machines (LPT algorithm). If all batches cannot be assigned to machines due to the constraints for tool life and available tool copies, go to Step 6. Otherwise, go to Step 5.

1) Select a batch with the maximum workload (processing requirement of a batch × minimum processing time) among the set of batches not allocated to a machine yet.
2) Assign the selected batch to the machine with the minimum workload allocated to it so far (ties are broken arbitrarily). If the batch cannot be assigned, constitute new batches for the operation by dividing its remaining processing requirement by $m' + 1$ (number of unassigned batches for the operation), and go to Step (1). Here, m' is the number of unassigned batches for the operation.

Step 5. If Z is improved, update Z and save the solution. Otherwise, go to Step 6.

Step 6. Let $m = m + 1$ and go to Step 2.

Although the initial solution obtained at the first stage can give a feasible solution, it can be improved further. The basic idea of the improvement stage is to remove an operation from the machine with the maximum workload and inset it to another machine. The procedure for the improvement stage is given below.

Procedure 2. (Improvement)

Step 1. Select machine m^* with the largest workload.

Step 2. For the selected machine m^*, do the following steps.

1) Among the unconsidered set of operations in machine m^*, i.e., $\{(i, j) \mid x_{ijm^*} = 1\}$, select the one with the largest processing time.
2) If there exists other machine m ($m \neq m^*$) to which the selected operation can be assigned, insert it to machine m and go to Step 1. Otherwise, remove the selected operation from the unconsidered set of operations in machine m^*. If the unconsidered set of operations is empty, stop. Otherwise, go to Step (1).

B. MULTIFIT based heuristic

This heuristic is the same as the LPT based one except that the batches constituted for a given number of machines to be assigned to each operation are allocated using the MULTIFIT algorithm, a multi-pass algorithm for the bin-packing problem. It makes repeated trials for batch assignments with different processing time capacities. The solution is the smallest capacity with which all the batches can be allocated to the machines and it is found by a bisection search method. In this study, the initial upper and lower bounds (UB and LB) were set as follows.

$$\sum_{i=1}^{I}\sum_{j=1}^{J_i}\max_m\{p_{ijm}\} \text{ and } \sum_{i=1}^{I}\sum_{j=1}^{J_i}\min_m\{p_{ijm}\}/M$$

In this algorithm, the LPT based heuristic explained earlier is used to assign operations to machines. The detailed procedure of the MULTIFIT algorithm (stage 1) is given below. Note that the improvement stage is the same as that of the LPT based heuristic.

Procedure 3. (Obtaining an initial solution: MULTIFIT)

Step 1. Let $Z = \infty$ and $m = 1$.

Step 2. If $m > M$, stop. Otherwise, go to Step 3.

Step 3. Do the following steps. (MULTIFIT algorithm).

1) Initialize the upper and the lower bounds on the capacities of machines.
2) If difference of the lower and the upper bounds is close enough, go to Step 4. Otherwise, set the machine capacities to be the midpoint of the bounds.
3) Allocate the batches to machines by the LPT based algorithm under value m. If all the batches can be assigned, let the current machine capacity be a new upper bound. Otherwise, let the current capacity be a new lower bound and go to (2).

Step 4. If Z is improved, update Z and save the solution and let $m = m + 1$ and go to Step 2.

IV. COMPUTATIONAL EXPERIMENTS

To show performances of the heuristics suggested in this paper, computational experiments were done on randomly generated test instances, and the results are reported in this section. Two performance measures were used: (a) percentage deviations from the optimal solution values; and (b) average CPU seconds. The optimal solutions were obtained by solving [P] using CPLEX 11.0, a commercial integer programming software package. The data generation method is omitted here due to the space limitation.

Test results are summarized in Table 1. As can be seen in the table, both algorithms can give near optimal solutions. Of the two heuristics, the MULTIFIT based one gave better solution than the LPT based one in overall average although no one dominates the other. In fact, the overall average gaps were 2.9% and 3.1% for the MULTIFIT and the LPT based heuristics, respectively. Also, the two heuristics were very fast. However, the CPLEX required much time although it can give the optimal solutions.

TABLE I. TEST RESULTS FOR HEURISTICS

NM[a]	NP[b]	Algorithms		CPLEX
		LPT based	MULTIFIT based	
3	10	2.2 (< 0.01)[c]	2.2 (0.01)	(4.59)
	15	3.1 (< 0.01)	3.0 (0.01)	(17.59)
5	10	3.4 (< 0.01)	3.0 (0.05)	(383.55)
	15	3.8 (< 0.01)	3.4 (0.05)	(1534.50)

a. Number of machines

b. Number of part types

c. Average percentage gap and CPU seconds (in parenthesis)

V. CONCLUSIONS

This paper considered a loading problem for system setup in flexible manufacturing systems with partially grouped unrelated machines and additional tooling constraints, i.e., available tool copies and tool life restrictions. An integer linear programming model is suggested for objective of minimizing the maximum workload. Then, two heuristics, consisting of obtaining an initial solution and improvement, were suggested based on the LPT and the MULTIFIT algorithm for the bin packing problem. The test result showed that the MULTIFIT based heuristic outperforms the LPT based one while requiring very short computation time.

One of important further researches is to combine the loading algorithm with the part type selection problem or scheduling problem. Since the part type selection and loading problems are interrelated with each other, the two problems may have to be solved simultaneously. In this case, the loading algorithm suggested in this paper can be used as a subroutine for an algorithm for part type selection.

ACKNOWLEDGEMENT

This work was supported by Ministry of Knowledge Economy (MKE) granted funded by Korea government (Grant Code: 10033895-2009-11). We also thank to Mr. Kang, K.-W., Mr. Jeong, T.-S., Mr. Lee J.-H. and Mr. Ko, H. for their helps in the computational test.

REFERENCES

[1] K. E. Stecke, "Formulation and solution of nonlinear integer production planning problem for flexible manufacturing systems" Management Science, Vol. 26, Mar. 1983, pp. 273-288, doi:10.1287/mnsc.29.3.273.

[2] D.-H. Lee and Y.-D. Kim, "Loading algorithms for flexible manufacturing systems with partially grouped machines" IIE Transactions, Vol. 32, Jan. 2000, pp. 33-47, doi:10.1023/A:1007651329917.

[3] Y.-D. Kim and C.A. Yano, "A heuristic approach for loading problems in flexible manufacturing systems" IIE Transactions, Vol. 25, Jan. 1993, pp. 26-39, doi:10.1080/07408179308964263.

[4] R. Swarnkar, and M. K. Tiwari, "Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing-based heuristic approach" Robotics and Computer Integrated Manufacturing, Vol. 20, Jun. 2004, pp. 199-209, doi:10.1016/j.rcim.2003.09. 001

[5] A. Kumar, Prakash, M. K. Tiwari, R. Shankar, and A. Baveja, "Solving machine-loading problem of a flexible manufacturing system with constraint-based genetic algorithm" European Journal of Operational Research, Vol. 175, Dec. 2006, pp. 1043-1069, doi:10.1016/ j.ejor.2005.06.025.

[6] S. Biswas and S.S. Mahapatra, "Modified particle swarm optimization for solving machine-loading problems in flexible manufacturing systems" International Journal of Advanced Manufacturing Technology, Vol. 39, Nov. 2008, pp. 931-942, doi:10.1007/s00170-007-1284-5.

[7] K. E. Stecke and N. Raman, "Production planning decisions in flexible manufacturing systems with random material flows" IIE Transactions, Vol. 26, Sep. 1994, pp. 2-17, doi:10.1023/A:1012290630540.

[8] A. Grieco, Q. Semeraro, and T. Tolio, "A review of different approaches to the FMS loading problem" International Journal of Flexible Manufacturing Systems, Vol. 13, Oct. 2001, pp. 361-384, doi:10.1023/A:1012290630540.