ISSN 1996-3386

Asian Journal of Industrial Engineering

Knowledgia
SCIENTIFIC
A Place to Publish Outstanding Research

http://knowledgiascientific.com

Asian Journal of Industrial Engineering 2 (1): 1-8, 2010
ISSN 1996-3386

A Heuristic Approach for Workload Balancing Problems

[1]Jonghwan Lee, [2]César O. Malavé, [1]Tae-Sung Kim and [1]Sae-Jae Lee
[1]Department of Industrial and Systems Engineering,
Kumoh National Institute of Technology, South Korea
[2]Department of Industrial and Systems Engineering,
Texas A and M University, College Station, Texas, USA

Abstract: The loading problem in a Cellular Manufacturing System (CMS) lies in the allocation of operations and associated cutting tools to machines for a given set of parts subject to capacity constraints. This research suggested a heuristic approach to the machine loading problem under the constraints of the workload and tool magazine capacity of each machine. This approach tried to reduce the maximum workload of the machines by partially grouping them. The processing time of the operation is different for each machine group, which is composed of the same identical machines; however, these machines can perform different sets of operations if tooled differently. The heuristic demonstrates the efficiency of allocating operations to each group and this problem is formulated as an integer linear problem. Performance of the suggested loading heuristics is tested by means of randomly generated tests. The result of this research, which is a well-balanced workload system, is obtained and partial grouping is a critical means of obtaining that goal. Partial grouping yields a more balanced workload because it entails the subdivision of demands into several batches.

Key words: Cellular manufacturing system, machine loading, workload balancing, partial grouping

INTRODUCTION

After product items and their quantities to be manufactured are determined by production planning, the next problem to be solved in production management is that of allocating the workloads to the existing production facilities for manufacturing these products. In general, the capacities (including human power) of the facilities are not infinite. Therefore, in order to actually perform production activities according to the production plan established, it is essential to adjust the workload for each of the facilities and workers in every time period so they do not exceed the given capacity. This decision is called machine loading.

After assigning an operation to multiple machines, a set of tools are loaded onto each machine that is required for that operation. This is one of the distinct characteristics of partial grouping. In other sorts of grouping, tools are loaded before operations are assigned, however, in partial grouping, necessary tools are loaded after operations allocated to each machine. It is this characteristic of partial grouping that makes each machine a virtual cell.

Corresponding Author: Jonghwan Lee, Department of Industrial and Systems Engineering, Kumoh National Institute of Technology, South Korea

In most cases in the field of production planning and scheduling, the processing time required to complete a specified operation is set as a constant. In most of the existing research about loading problems with respect to partial grouping, the machine loading models were constructed under the assumption that the processing time is a constant. In practical situations, however, it is possible to vary the processing times by actively changing manufacturing conditions, especially machining speeds. In these cases, some modifications must be made to production planning and scheduling models. To be useful, those models require a new type of heuristic that allows for variation in processing times. This research is concerned with machine loading models that have variable processing times.

The most significant contributions of the research described in this research are (1) the development of a good heuristic in a loading problem with variable processing time for each cluster; (2) assigning the operations into clusters. (3) the implementation of different heuristics, according to phase. Operations are assigned to clusters while each cluster has a different processing time. (4) the attempt to address the loading problem with respect to each machine's capacity and workload limit, which impact loading problem performance.

There is extensive literature on assembly line and workload balancing in job shops and Cellular Manufacturing System (CMS). Balancing is appropriate for flexible assembly systems as well as automated transfer lines. Stecke and Solberg (1981) employed loading and control policies for a flexible manufacturing system and defined loading and control methods that significantly improve system production rates. Stecke (1986) considered various operation assignment objectives appropriate in FMS and presented a hierarchical framework for considering these objectives. Berrada and Stecke (1986) applied a branch and bound algorithm to solve the workload balancing problem for all machines when each machine's processing time is different. A new branch and bound algorithm was developed, based on the work of Berrada and Stecke (1986). This new algorithm was developed to maximize the expected production rate (throughput) of the system and to ensure that actual workload allocation is commensurate with the continuous workload allocation that maximizes throughput. There are various algorithms for the identical processor minimum-makespan scheduling problem and for some of them the worst case performance ratios are known. The LPT algorithm has a worst case performance ratio of 4/3-1/3 m, where, m is the number of processors (machines) and the Multifit algorithm has a worst case performance ratio of $1.2+(1/2)^k$, where, k is the number of iterations in the algorithm (Coffman *et al*., 1978; Friesen, 1984; Graham, 1969). Lee and Kim (2000) implemented several loading algorithms for flexible-manufacturing systems with partially grouped machines. They formulated the loading problem by means of integer programming and primarily utilized LPT and Multifit algorithms. They described two means of addressing this problem. One approach was direct, whereas, the other decomposed the problem into the operation assignment problem and the workload allocation problem. Both approaches implemented LPT and Multifit algorithms, yet a comparison of the results demonstrated that decomposition methods are better that direct ones. Additionally, simulation experiments demonstrated that partial grouping loading plans yielded significantly better performance than total grouping loading plans. Chen and Askin (1990) performed heuristics, based on the separate evaluation of five objectives: workload balance, volume of inter-machine part movement, routing flexibility, tool investment and maximum machine utilization. Choi and Lee (1998) developed heuristic procedures with the two-part objective of minimizing workload imbalances and maximizing system throughput. Their solution hinged on the rejection factor and virtual total processing time. Shanthikumar and Stecke (1986) showed that maintaining balanced workloads on each machine over time stochastically minimizes work-in-process inventory requirements for FMS that contain only one machine in a group. Stecke and Morin (1985) have shown that if each operation is assigned to only one machine, balancing the workload of each machine maximizes expected production by using symmetric mathematical programming. Nagarjuna *et al.* (2006) proposed a heuristic to minimize the system unbalance based on multi-stage programming approach. Ponnambalam and Kiat (2008) also tried to minimize system unbalance and additionally maximizing system throughput by using of a particle swarm optimization algorithm.

The aim of this research was to investigate the formulation of the loading problem as an integer programming problem, to develop a solution algorithm based on the formulation of the problem and to test solution methodologies. Eventually, these procedures will minimize the maximum workload for each machine.

MODEL DEVELOPMENT

The purpose of this problem is to assign each operation to machines. In order to assign operations to machines, those operations must be assigned to cluster first and then they can be assigned to machines in each cluster. The machines used for these CMS have automatic tool changers and a tool magazine of a limited capacity and can work limited time for each day. To execute an operation, one or more tools are required and each tool requires one or more slots in the tool magazine. Another significant feature of this system is tool sharing or tool commonality. In other words, several operations may share the same tools in the system. Operations require several tools: yet if different operations are allocated to the same cluster or different machines use the same tool, then duplicated tools are not assigned to the same cluster or machine.

If operations require different processing times, they are considered different operations even though they are the same type. For instance, drilling operations for different part types are treated as different operations if their processing times are different-even though they require the same set of tools. The loading problem allocates operations and associated tools to machines in order to minimize the maximum workload of the machines subject to tool magazine capacity and workload capacity constraints. An integer linear problem provides a clear description of this loading problem. The following notations are used in its formulation.

The notation used in problem formulation is presented:

I = Index for operation, $i \in I$
j = Index for machine, $j \in J$
t = Index for tools, $t \in T$
k = Index for machine type, $k \in K$
c = Index for machine cluster, $c \in C$
p_i = Processing time of operation i
p_{ki} = Processing time of operation i for machine type k
C = Capacity of a tool magazine of a machine cluster j in cluster c
W_{cj} = The workload for a machine j in the cluster
W_c = The maximum workload allowed for a cluster c
D_i = Demands of operation i
s_t = Number of tool slots needed for too t
b_{ic} = Batches for operation i that is assigned to machine clustec
$|G|$ = Number of cluster

$$a_{it} = \begin{cases} 1 & \text{if operation i requires tool t} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ic} = \begin{cases} 1 & \text{if operation i assigned to machine cluster c} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ic} = \begin{cases} 1 & \text{if tool t is assigned to machine cluster c,} \\ 0 & \text{otherwise.} \end{cases}$$

Formulation

Minimize Z, subject to:

$$\sum_{k \in K} \sum_{i \in I} p_{ki} b_{ic} \leq Z \qquad \forall\, c \tag{1}$$

$$W_c = \sum_{j \in J} W_{cj} \qquad \forall c \tag{2}$$

$$1 \leq \sum_{c \in G} x_{cj} \leq |G| \qquad \forall c, j \tag{3}$$

$$b_{ic} \leq D_i x_{ic} \qquad \forall i, c \tag{4}$$

$$\sum_{c \in C} b_{ic} = D_i \qquad \forall\, i \tag{5}$$

$$a_{it} x_{ic} \leq y_{tc} \qquad \forall i, c, t \tag{6}$$

$$\sum_{t \in T} s_t y_{tc} \qquad \forall c \tag{7}$$

$$p_{ki}, b_{ic} \geq 0 \qquad \forall i, k, c \tag{8}$$

$$x_{ic}, y_{tc} \geq \{0,1\} \qquad \forall\, i, c, t \tag{9}$$

In the formulation, constraint set (1) explains that operations can be assigned to any machine cluster and workload for each cluster is restricted. Constraint set (2) defines the sum of all workloads of the machines in each cluster. The number of clusters allocated to each operation should not be greater than the total number of clusters is denoted in constraint set (3). Constraint set (4) shows the relationship between b_{ic} and x_{ic}, which is the batches of operation i can be allocated to cluster c (b_{ic}) only if x_{ic}=1. Constraint set (5) shows that the sum of all batches to be assigned to each cluster in each operation is the demand of each operation. Constraint set (6) illustrates that the required tools for the operation to be loaded onto each cluster where the operation is allocated. Constraint sets (7) bound the number of parts that can be processed on a cluster and a machine within a cluster.

PROPOSED RESEARCH PROCEDURE

Four heuristics and a Multifit algorithm for each heuristic will be implemented in this research. The first Heuristic selects the maximum batch workload for each cluster and then selects the minimum batch workload among them. The second Heuristic selects the minimum batch workload value for each cluster and then selects the minimum batch workload value among them instead of selecting the maximum batch workload value of the minimum from the cluster types. The third Heuristic selects the maximum batch workload value for each cluster and then selects the maximum batch workload value from among them. The fourth Heuristic selects the minimum batch workload value in each cluster and then chooses the maximum batch value among them. By means of these trials, we can determine which of the four Heuristics provides the best performance. The modified Multifit algorithm will be applied with each Heuristic. W_c and C_c are control parameters for the heuristic. When assigning operations, we use W_c and C_c to limit the operation aggregation process. As W_c increases, fewer but longer operations will be fed to the solution procedure. Material handling should decline but workload balancing may become more difficult in the cluster.

We will use the term $\Delta\lambda_{ic}$ as a dynamic variable to indicate the number of machine tool slots that must be added to cluster type cto perform operation i. This term is dynamic in the sense that it depends on which tools have already been assigned to c. For example, if two operations use the same tool and the first operation is assigned to clusters, the second operation must then use additional tool slots.

And the term $\Delta\eta_{ic}$ will be used as a dynamic variable for denoting the workload limit for each cluster. There is a workload limit for each machine and the total workload limit for the machines within each cluster is the workload limit for that cluster. This workload limit value is decreased as it is when assigning operations to clusters. If the workload limit for each cluster has the negative value, then the system is infeasible.

The tool magazine capacity for each cluster defines the system in the same way. If the system is infeasible, then the system stopped and other variables are generated for it. If these problems happen repeatedly, then tool magazine capacities and workload limits for each machine must be considered. Flexible workload limits maximize the probability that all of the operations can be assigned to clusters; however, they also lower utilization. The main factor that we need to consider is the workload, because the objective is minimize the maximum workload so that we can minimize the lead time with these results.

General Procedure for Heuristic

Step 1: Initialize maximum workload and maximum tool capacity

Step 2: Make batches for each operation by dividing demands of an operation by the number of clusters (Batches must be integers, so if demand=10 and the number of machines is 3, then the batches are 3, 3, 4)

Step 3: Initialize batch workload by multiplying batch and processing time for each operation

Step 4: In each cluster, select the minimum batch workload and then select the maximum batch workload from those previously selected

Step 5: For the operations that have been selected, select operations that have the same batch number for each cluster

Step 6: For all selected operations for each cluster, select if maximum workload limit selected batch workload ($\Delta\eta_{ic}$)> 0 and maximum tool capacity-the number of tool ($\Delta\lambda_{ic}$) > 0

Step 7: Select maximum value of maximum workload limit – selected batch workload ($\Delta\eta_{ic}$)

Step 8: Assign operation to the cluster that has the largest remaining workload capacity if tools were assigned previously, then do not allocate again. Update maximum workload limit and maximum tool capacity

Step 9: Repeat until all operations are allocated

Table 1: Selection rule for operation assignments

Heuristics	Between cluster	Among selected ones	Assign operations to clusters
I	Maximum workload	Minimum workload	Largest remaining cluster
II	Minimum workload	Minimum workload	Largest remaining cluster
III	Maximum workload	Maximum workload	Largest remaining cluster
IV	Minimum workload	Maximum workload	Largest remaining cluster

Table 2: Generating loading problem data sets

Parameters	Range of values
Demand	Between 5 and 30
The processing time	Between 1 and 30
The No. of tools for each operations	Between 5 and 10
The No. of tool slots needed for each tool	1, with probability 0.7
	2, with probability 0.1
	3, with probability 0.2
Total No. of tools used in this tests	80

Table 3: Configurations for running loading problems

No. of cluster	The No. of machine	The No. of operation	Tool capacity for cluster and machine	Workload limit for the cluster	Workload limit for the machine
3	4	90	110, 140	9200	2300
3	6	140	110, 140	13800	2300
3	8	190	110, 140	18400	2300
4	4	140	110, 140	9200	2300
4	6	200	110, 140	13800	2300
4	8	280	110, 140	18400	2300
5	4	190	110, 140	9200	2300
5	6	280	110, 140	13800	2300
5	8	360	110, 140	18400	2300

Table 1 presents configurations for each Heuristics. Unfortunately, it is extremely difficult to obtain actual data for a wide variety of systems because most are proprietary. In order to overcome this limitation, test problems have been generated randomly to ensure that the resulting data represent real systems relatively well. Test case data was randomly generated using the parameters and test levels provided in Table 2 and 3.

Four Heuristics are shown in the first column. The second column shows how to select between clusters. The third column describes the means for selecting operations in column two. The last column describes the means for allocating operations to each cluster. The Multifit algorithm will be implemented for each Heuristic case.

RESULTS

This research has not been done in any other previous research, so we decided to show the performance of result by comparing the outcome with performance ratio, which is lower bound. Solutions from the algorithms were compared with each other using the performance ratio as an index; this is the ratio to a lower bound, i.e.,

$$\sum (p_i \cdot D_i)/|C|$$

After implementing all of the Heuristics, a relative performance ratio, which is defined by as $[(H_h\text{-}H_B)/H_h]\times 100$, will be shown in order to evaluate the results. H_B is the best performance ratio and H_h is the performance ratio from the Heuristic.

Table 4: Relative performance in cluster 3

No. of machine	No. of operation	No. of tool	Cluster work-loadlimit	Heuristics I	II	III	IV
4	90	110	9200	0.1023	0.0584	0.0729	0.1102
4	90	140	9200	0.0913	0.0494	0.0418	0.0889
6	140	110	13800	0.1027	0.0654	0.0522	0.1049
6	140	140	13800	0.1069	0.0577	0.0406	0.1051
8	190	110	18400	0.0827	0.0533	0.0683	0.0868
8	190	140	18400	0.0818	0.0399	0.0683	0.0802
Average				0.0946	0.0540	0.0526	0.0960
No. of machine	No. of operation	No. of tool	Cluster work-loadlimit	Heuristics I	II	III	IV
4	90	110	9200	0.1063	0.0621	0.0762	0.1137
4	90	140	9200	0.0945	0.0527	0.0453	0.0922
6	140	110	13800	0.1062	0.0688	0.0551	0.1085
6	140	140	13800	0.1095	0.0610	0.0432	0.1083
8	190	110	18400	0.0855	0.0566	0.0716	0.0900
8	190	140	18400	0.0850	0.0427	0.0431	0.0834
Average				0.0946	0.0540	0.0526	0.0978

Table 5: Relative performance in cluster 4

No. of machine	No. of operation	No. of tool	Cluster work-loadlimit	Heuristics I	II	III	IV
4	90	110	9200	0.0788	0.0505	0.0515	0.0795
4	90	140	9200	0.0852	0.0468	0.0348	0.0847
6	140	110	13800	0.0730	0.0566	0.0447	0.0706
6	140	140	13800	0.0694	0.0458	0.0500	0.0680
8	190	110	18400	0.0418	0.0411	0.0257	0.0396
8	190	140	18400	0.0512	0.0373	0.0325	0.0511
Average				0.0666	0.0464	0.0399	0.0656
No. of machine	No. of operation	No. of tool	Cluster work-loadlimit	Heuristics I	II	III	IV
4	90	110	9200	0.0821	0.0531	0.0549	0.0825
4	90	140	9200	0.0882	0.0506	0.0382	0.0881
6	140	110	13800	0.0757	0.0602	0.0482	0.0734
6	140	140	13800	0.0721	0.0492	0.0528	0.0709
8	190	110	18400	0.0451	0.0443	0.0292	0.0431
8	190	140	18400	0.0543	0.0402	0.0350	0.0542
Average				0.0696	0.0496	0.0432	0.0687

Given these results of loading problem computational experiments, one can infer that relative performance is good and can be applied in real factory problem. Between relative performance ratio for 8 heuristics in each cluster in the Table 4-6, Heuristic III performed well. The result of Heuristic III shows very low performance ratio, which is close to 0. It means that the outcome of this Heuristic III is very close to lower bound. We expect this result will reduce lead times and operating costs. Furthermore, such outputs may be obtained in a reasonable running time.

In analyzing this loading problem, this research has made several contributions. The first was the identification and formal modeling of the loading problem. The integer linear programming models presented were extensions of known models for the loading problem. A second contribution was the ability to deal with different processing times in different clusters for the same operation. This reflects the scenario in which an operation can be processed on machines of different types.

Table 6: Relative performance in cluster 5

No. of machine	No. of operation	No. of tool	Cluster work-loadlimit	Heuristics I	II	III	IV
4	90	110	9200	0.0692	0.0605	0.0543	0.0639
4	90	140	9200	0.0576	0.0516	0.0385	0.0650
6	140	110	13800	0.0450	0.0532	0.0474	0.0498
6	140	140	13800	0.0417	0.0518	0.0324	0.0472
8	190	110	18400	0.0470	0.0681	0.0486	0.0493
8	190	140	18400	0.0396	0.0643	0.0308	0.0373
Average			0.0464	0.0582	0.0420	0.0521	

No. of machine	No. of operation	No. of tool	Cluster work-loadlimit	Heuristics I	II	III	IV
4	90	110	9200	0.0731	0.0636	0.0585	0.0676
4	90	140	9200	0.0612	0.0553	0.0432	0.0689
6	140	110	13800	0.0483	0.0569	0.0515	0.0539
6	140	140	13800	0.0450	0.0548	0.0360	0.0512
8	190	110	18400	0.0506	0.0720	0.0524	0.0530
8	190	140	18400	0.0435	0.0680	0.0340	0.0410
Average				0.0496	0.0618	0.0459	0.0560

ACKNOWLEDGMENT

This study was supported by Research Fund, Kumoh National Institute of Technology.

REFERENCES

Berrada, M. and K.E. Stecke, 1986. A branch and bound approach for machine load balancing in flexible manufacturing systems. Manage. Sci., 32: 1316-1335.

Chen, Y.J. and R.G. Askin, 1990. A multiobjective evaluation of flexible manufacturing system loading system heuristics. Int. J. Prod. Res., 28: 895-911.

Choi, S.H. and J.S. Lee, 1998. A heuristic approach to machine loading problem in non-preemptive flexible manufacturing systems. Int. J. Indus. Eng., 5: 105-116.

Coffman, E.G.Jr., M.R. Garey and D.S. Johnson, 1978. An application of bin-packing to multiprocessor scheduling. SIAM J. Comput., 7: 1-17.

Friesen, D.K., 1984. Tighter bounds for the multifit processor scheduling algorithm. SIAM J. Comput., 13: 170-181.

Graham, R.L., 1969. Bounds on multiprocessor timing anomalies. SIAM J. Applied Math., 17: 416-429.

Lee, D.H. and Y.D. Kim, 2000. Loading algorithms for flexible manufacturing systems with partially grouped machines. IIE Transactions, 32: 33-47.

Nagarjuna, N., O. Mahesg and K. Rajagopal, 2006. A heuristic based on multi-stage programming approach for machine-loading problem in a flexible manufacturing system. Robotics Comput. Integrated Manuf., 22: 342-352.

Ponnambalam, S.G. and L.S. Kiat, 2008. Solving machine loading problem in flexible manufacturing systems using particle swarm optimization. Eng. Technol., 39: 14-19.

Shanthikumar, J.G. and K.E. Stecke, 1986. Reducing work-in-process inventory in certain types of flexible manufacturing systems. Eur. J. Operational Res., 26: 266-271.

Stecke, K.E. and J.J. Solberg, 1981. Loading and control policies for a flexible manufacturing system. Int. J. Prod. Res., 19: 481-490.

Stecke, K.E. and T.L. Morin, 1985. The optimality of balancing workloads in certain types of flexible manufacturing systems. Eur. J. Operat. Res., 20: 68-82.

Stecke, K.E., 1986. A hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems. Eur. J. Operat. Res., 24: 369-378.