Tool Planning Problem in Flexible Manufacturing Cells
- FMC에서의 공구계획문제 -

Na, Yoon Kyoon*
나 윤 균

요 지

공구계획 문제는 FMC의 설계 단계에서 고려되어야만 하는 중요한 요소 중의 하나이다. 본 연구에서는 FMC에서 흔히 사용되는 머쉬닝센터를 대상으로 공구이동정책을 사용할 때 공구의 이동을 최소화할 수 있는 공구계획에 그 목적을 두고 있다. 공구매거진의 제한된 공구수용능력과 머쉬닝센터 간의 공구이동량을 최소화하여 작업의 공구대기시간을 감소시킴으로써 기계의 이용율을 높이고 생산량을 증대시킬 수 있다. 총공구 수가 주어진 조건하에서 공구의 이동량을 최소화하면서 각 공구의 숫자를 결정하는 수학적 모델을 개발하여 효율적인 해법을 제시하였으며, 실험결과 짧은 시간에 우수한 해를 제공하는 것이 입증되었다.

1. Introduction

In flexible manufacturing cells (FMC) where a large number of different cutting tool types are required, a single machining center may not hold all tool types required for the shop operations due to the tool magazine capacity. As a result, tools are partitioned to several machining centers. So if a job does not find all required tools on a machining center, it has to visit more than one machining center. This approach is common in many metal cutting industry. However, every time a job visits different machining centers, repositioning of the job at a reference point is required and may cause the loss of cutting precision. An altenate approach proposed by Han et al.[6] is to let the job stay at a machining center and borrow those required tools from other machining centers. The approach of tool borrowing is based on the assumption that each machining center has the capability of performing all operations required of a given job if proper tools are provided. When a job remains at one machining center, there is no need of repositioning and consequently cutting precision can be maintained. Also work-in-process inventories would be reduced if induction could be controlled properly.

However, tool borrowings incur tool waiting time because if not all the tools required for a block of operations for the job are ready, the cutting operations can not be started. Those tools to borrow from other machining centers or from a tool crib can not be readily extracted from their magazines until the completion of the current block of operations, due to safety reasons. As a means to reduce tool wating time, commonly used tools are duplicated and assigned to different machining centers, thereby reducing the need for tool

* Dept. of Ind. Eng., Univ. of Suwon

borrowings. Tool borrowings can also be reduced by proper grouping of tool types on each machining center and proper job assignment to machining centers. However, the job assignment which reduces tool borrowings may not result in balanced workload distribution among machining centers. Since our ultimate objective is throughput rate maximization, we would not want extremely unbalanced workload on machining centers only to achieve the reduction in tool wating time and reduced work-in-process inventories.

These two objectives, the reduction in tool wating time caused by tool borrowings and balancing of workload, however, are often contradictory. Usually, minimizing the amount of tool wating time is accompanied with the increase in workload imbalance among machining centers, and vice versa. In this paper, tool changes due to tool wear are not considered because tool wear is not affected by tool grouping of job assignment to machining centers.

A limited amount of research work has been reported in tool loading in flexible manufacturing systems. Berrada and Stecke [3] developed a solution method for the objective of workload balance when jobs are routed from machining center to machining center. Ammons et al.[1] proposed a workcenter loading problem in flexible assembly where the workload balance and part movements were the main concern. Tang [11] has proposed a job scheduling model which minimizes the number of tool changes on a single machining center.

In this paper, as a vehicle to the ultimate objective of throughput rate maximization, we want to reduce the amount of tool borrowings and at the same time balance the workload of each machining center. The reduction in tool traffic and the balance of workload are achieved through the determination of the optimum number of tool copies of each tool type when the total number of tool copies is limited, through the proper partition of the tool set into multiple machining centers, and through the proper assignment of jobs to each machining center.

The plan of this paper is as follows. In section 2, assumptions and notation are introduced. Section 3 shows the problem formulation as a nonlinear integer programming problem, and section 4 describes an approximate solution method. Conclusions are made in section 5.

2. Assumptions and Notation

2.1 Assumptions

It is assumed that at the outset of a production horizon, a job set to be completed by the FMC is known. The problem setting is viewed as static in the sense that those jobs in the job set would not compete for the priority for the early completion. The followings are further assumptions :

(1) All machining centers are identical, and can perform operations on any job in the job set if necessary tools are provided;

(2) A block of operations cannot be started until all the required tools are placed in the magazine;

(3) Tools can not be extracted from the tool magazine if the machining center is in cutting operation;

(4) Each magazine has a limited capacity of holding tools; and

(5) The tools borrowed are returned upon the completion of operation.

2.2 Notation

The following notation for indices, parameters, and decision variables is listed here for reading convenience.

machining center index $i=1,2,\cdots,m$

part index $j=1,2,\cdots,n$

tool index $t=1,2,\cdots,\ell$

a_{jt} = 1 if part j requies tool t

0 otherwise

c_t = number of tool copies for tool type t

r_j = number of tools required by part type j ($\sum_{t=1}^{\ell} a_{jt}$)

p_j = processing time of part type j

s_i = capacity of magazine at machining center i

b = exact balance of workload

α = workload imbalance factor

TC = total number of tool copies allowed

J = job set

M = machining center set

T = tool set

W(i) = workload of machining center i

U = upper workload imbalance limit $(b*(1+\alpha))$

L = lower workload imbalance limit $(b*(1+\alpha))$

x_{it} = 1 if tool t is assigned to the magazine of machining center i

0 otherwise

y_{ij} = 1 if part j is assigned to machining center i

0 otherwise

If there are as many tool copies for a given tool type as the number of machining centers, each machining center need not borrow the tool type from other machining center. Thus, when $TC \le m*\ell$, it is obviously advantageous to have at least m tool copies for each tool type, ie, $c_t \ge m$. Therefore, we restrict out interest to the case when $TC < m*\ell$, and there can be following two cases:

Case 1. The total number of tool copies is less than or equal to the total magazine capacity, ie, $TC \le \sum_{i=1}^{m} S_i$.

Case 2. The total number of tool copies is greater than the total magazine capacity, ie, $TC > \sum_{i=1}^{m} S_i$.

For Case 2, all tool copies cannot be held in tool magazines and the rest is kept in a dummy magazine with capacity equal to $(TC-\sum_{i=1}^{m} S_i)$.

Tools assigned to the dummy magazine will represent those tools assigned to a tool crib.

3. Mathematical Formulation

The decision of tool copies for each tool type and their assignment to different machining centers depend upon the nature and contents of the job set to be completed. However, the contents of a job set changes from production period to period. Therefore, at least for the tool planning problem, production schedule should be established over a planning horizon, which may consist of a number of production periods.

P1 : minimize $$\sum_{i=1}^{m}\sum_{j=1}^{n}\left(r_j - \sum_{t=1}^{l} a_{jt} \times x_{it}\right) \times y_{ij} \qquad (1)$$

subject to $$\sum_{t=1}^{l}\sum_{i=1}^{m} x_{it} \leq TC \qquad (2)$$

$$\sum_{i=1}^{m} x_{it} \geq 1,\ t=1,2,\cdots,l \qquad (3)$$

$$\sum_{t=1}^{l} x_{it} \leq s_i,\ i=1,2,\cdots,m \qquad (4)$$

$$\sum_{i=1}^{m} y_{ij} = 1,\ j=1,2,\cdots,n \qquad (5)$$

$$b(1-\alpha) \leq \sum_{j=1}^{n} p_j \times y_{ij} \leq b(1+\alpha),\ i=1,2,\cdots,m \qquad (6)$$

$$x_{it}=0 \text{ or } 1, \text{ for all } i, t \qquad (7)$$

$$y_{ij}=0 \text{ or } 1, \text{ for all } i, j \qquad (8)$$

3.1 Objective Function

The main concern is to determine tool copies for each tool type and partition them to machining centers in such a way that the amount of tool borrowings between machining centers is minimized. The number of tool types required for all operations of part j, r_j, is given by :

$$r_j = \sum_{t=1}^{l} a_{jt}.$$

The number of tool types currently available at the magazine of machining center i for the operations of job j, e_{ij}, is given by : $e_{ij} = \sum_{t=1}^{l} a_{jt} \times x_{it}$.

Thus, the number of tools to be borrowed if job j is assigned to machining center i is equal to $(r_j - e_{ij})$. The objective function is to minimize the total number of tool borrowings, and expressed as equation (1).

3.2 Constraints

There are two sets of constraints ; one for tool loading (x variables) and the other for job assignment (y variables). The constraint set on the tool assignment includes the following considerations :

(1) the limitation or total tool copies allowed,

(2) at least one copy of each tool type in the tool set, and

(3) the limitation of magazine capacity.

The first constraint implies that the number of tool copies assigned to magazines of machining centers should not exceed the total tool copies allowed, which is given by equation (2). The second constraint of providing at least one copy for each tool type is given by equation (3). The third constraint, equation (4), is to reflect that the number of tools assigned to the tool magazine of a machining center cannot exceed its magazine capacity. The constraints specified by equations (2),(3), and (4) are for tool loading among machining centers.

For job assignment, there are two constraints. The first constraint, equation (5), ensures that each job is assigned to only one machining center. The second constraint is for workload balance. Ideal workload balance, b, can be expressed as b= ($\sum_{j=1}^{n} p_j$) / m, where the right hand side implies the total workload divided by the number of machining centers. In reality, it might be hard to expect exact balance. Therefore, acceptable workload balance is described using imbalance factor, α, which is between 0 and 1. The lower workload imbalance limit is set by b $\times(1-\alpha)$, and the upper workload imbalance limit by b$\times(1+\alpha)$. Thus, the workload assigned to machining center, i, should be within the upper and the lower workload imbalance limits as specified by equation (6).

The objective function of P1 is nonlinear and decision variables are integers. As a nonlinear integer programming problem, it would be difficult to solve fairly large sized problems. Therefore, an approximate solution procedure is developed which will be explained in the next section.

4. Solution Procedure

The nonlinear objective function can be linearized through introducing linear constraints in the constraint set. However, the linearization is accompanied with an exponential increase in the problem size. Thus, making use of the special structure of this problem, an approximate solution procedure has been developed.

The proposed solution procedure partitions the original problem into two subproblems ; one for tool loading problem and the other for job assignment problem. With a known solution of job assignment, the problem is reduced to tool loading problem with constraints consisting of equations (2), (3), (4), and (7), and with a known solution of tool loading the problem with constraints consisting of equations (5), (6), and (8). Then the two subproblems are solved iteratively untill some convergence is obtained as explained next.

4.1 Decomposition to Subproblems

With known values of y variables, $\hat{y}$, which satisfy equations (5), (6), and (8), the objective function becomes a linear function of x variables and constraint set consisting of equations (2), (3), (4), and (7), shown as follows :

S1 :

$$\text{minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n}\left(r_j - \sum_{t=1}^{l} a_{jt} * x_{it}\right) * \hat{y}_{ij}$$

Subject to $\sum_{t=1}^{l}\sum_{i=1}^{m} x_{it} \leq TC$

$$\sum_{i=1}^{m} x_{it} \geq 1, \; t=1,2,\cdots,l$$

$$\sum_{t=1}^{l} x_{it} \leq s_i, \; i=1,2,\cdots,m$$

$$x_{it} = 0 \; or \; 1, \; \text{for } all \; i,t$$

Let A denote the matrix of the constraint set of the S1. Then, A is totally unimodular and S1 can be solved as linear programming problem (see Papadimitriou and Steiglitz [9]). Thus, the integrality constraint can be relaxed and replaced with the following continuous constraint ; $0 \leq x_{it} \leq 1$, for all i, t.

Similarly, with the solution, from S1, P1 will have a linear objective funtion in y variables with constraint set consisting of equations (5), (6), and (8), shown as follows :

S2 :

minimize $\sum_{i=1}^{m}\sum_{j=1}^{n}\left(r_j - \sum_{t=1}^{l} a_{jt} * \hat{x}_{it}\right) * y_{ij}$

subject to $\sum_{i=1}^{m} y_{ij} = 1, \; j=1,2,\cdots,n$

$$b(1-a) \leq \sum_{j=1}^{n} p_j * y_{ij} \leq b(1+a), \; i=1,2,\cdots m$$

$$y_{ij}=0 \; or \; 1, \; \text{for } all \; i,j$$

S2, however, is an integer programming problem because integrality constraint imposed on y_{ij} variables cannot be removed.

4.2 Approximate Solution Method

The solution procedure starts with solving the job assignment first. The initial job with the longest processing time to the machining center with the minimum workload until all jobs are assigned. With known values of y variables, S1 can be solved exactly with a linear programming technique. However, S2 is an integer programming problem which requires much effort to solve exactly. Since the whole solution procedure is an approximate one, S2 is solved using a heuristic. The heuristic assigns a job to the machining center at which the least number of tool borrowings becomes necessary for the job, while maintaining the workload of each machining center between the lower and the upper workload imbalance limit.

Let d_{ij} denote the coefficient of y_{ij} in the objective function of S2. The heuristic for S2 starts with the machining center with the minimum workload and select a job which will cause the minimum number of tool borrowings from other machining centers. For instance, if machining center i' has the minimum workload currently, then job j' which has the following property is selected and assigned to machining center i' : $d_{i'j'}=\mathrm{Min}_{j\in J'}\{d_{i'j}\}$, where J' denotes the set of jobs which are not yet assigned. Repeat this procedure until the workload of each machining center exceeds the minimumn imbalance limit. After exceeding the minimum imbalancelimit for each machining center, job j′ is paired up with a machining center i′ if the assignment causes the minimum number of tool borrowings, described as follows : $d_{i'j'}=\mathrm{Min}_{j\in List}\{d_{ij}\}$, where List={(i,j) | j∈J′, i∈M}. If the assignment of job j′ to machining center i′ does not cause the workload of i′ to exceed the upper workload imbalance limit, then job j′ is assigned to machining center i′. Otherwise, the pair is removed from List in order to avoid unnecessary calculation at later iterations. Repeat this procedure until all jobs are assigned or List becomes empty. If List becomes empty before all jobs are assigned, no feasible solution exists which satisfies the workload imbalance limit. Once the problem is solved, the number of tool copies for each tool type is determined by $\sum_{i=1}^{m} x_{i_t}$.

The above procedure is put into an algorithmic form as follows :

Step 0. Use LPT(Longest Processing Time) rule to find an initial solution of y_{ij}^{0}.

z_1^n : objective function value of S1 at iteration n

z_2^n : objective function value of S2 at iteration n

x_{it}^n : solution of S1 at iteration n

y_{ij}^n : solution of S2 at iteration n

ε : small integer value

$z_2^0 \leftarrow 0$

$n \leftarrow 1$

Step 1. With $y_{ij}=y_{ij}^{n-1}$, solve S1 to find x_{it}^n and z_1^n. If $(z_2^{n-1}-z_1^n)<\varepsilon$, then go to step 4with the solution of y_{ij}^{n-1} and x_{it}^n. Otherwise, go to step 2.

Step 2. J←{Parts}

M←{Machining Centers}

W(i)←0 for all i

$z_2^n \leftarrow 0$

$$d_{ij} \leftarrow \sum_{i=1}^{m}\sum_{j=1}^{n}(r_j - \sum_{t=1}^{L} a_{i_t}\times x_{i_t})y_{i_t}$$

While not all W(i)<L do

i′ : $W(i)=\mathrm{Min}_{i\in m}\{W(i)\}$

$y_{i'j'} \leftarrow 1$

$z_2^n \leftarrow z_2^n + d_{i'j'}$

J←J-{j′}

W(i′)←W(i″)+pi′

Endwhile

List←{(i,j) | i∈ M ,j ∈ J}

While j≠Φ and List ≠Φ do

(i′,j′):d = $\mathrm{Min}_{(i,j)\in List}\{d_{ij}\}$

If $W(i')+p_{j'}<U$, then

$y_{i'j'} \rightarrow 1$

$z_2^n \rightarrow z_2^n + d_{i'j'}$

J← J - {j′}

W(i′)← W(i′) + p_i

else
  List ← List + {(i′,j′)}
 Endif
Endwhile
Step 3. If j=Φ and $z_1^n - z_2^n > \varepsilon$, then go to step 1.
Otherwise, go to step 4 with the solution of y_{ij}^{n-1} and x_{it}^n.
(Computation of Tool Copies)
Step 4. $c_t = \sum_{i=1}^{m} x_{it}$, for all t=1,2,⋯, ℓ

That the objective function values at each iteration do not increase if S1 and S2 are solved exactly (see Han el al.[6]). However, since S2 is not solved exactly in the solution procedure, the solution value of S2 could get greater than that of S1 at the same iteration. Therefore, if the proposed algorithm is made to stop when the desrease of tool movements at each step is less than a certain amount, ε,the algorithm will automatically stop whenever any deterioration is encountered or convergence becomes slow.

5. Experiments

Since P1 is a nonlinear integer programming problem, the computational burden for an exact solution will be enormously heavy, and thus an approximating solution procedure was proposed. A heuristic rule was also developed as a quicker solution method. Since there is no previous work to be compared, the proposed solution procedure and the heuristic rule are compared to each other.

Two scenario problem sets were randomly generated for the experiment of the proposed solution procedure and the heuristic rule. Scenario 1 consists of 20 tool types, 2 machining centers, and 40 or 60 parts; whereas scenario 2 consists of 30 tool types, 3 machining centers, and 60 or 90 parts. Scenario 2 consists of bigger problems than scenario 1 in order to discern the effect of problem size on the performance. Table 1 compares the two scenarios.

The total number of tool copies allowed is specified by the tool duplication factor, γ,and the number of tool types, ℓ,as follows:

$$TC=(1+\gamma)\,\ell \qquad (9)$$

For each scenario, three levels of tool duplication factor were tested, 0.3, 0.5 and 0.7. With γ = 0.3, scenario 1 problems were allowed to duplicate 30% of the total number of tool types, that is, 6 tool copies, and TC = 26. For each scenario problems set, two different part volumes were tested in order to discern the effect of part volume; for scenario one, 40 and 60 parts, and for scenario two, 60 and 90 parts. For each tool duplication factor level and part volume, 20 problems were tested. The workload imbalance factor level is fixed at 0.2 for all problems.

The processing time for each part was randomly selected from a uniform distribution, U[1,10]. The number of tool types required for a part was selected from a uniform distribution, U[3,7] and the tool types required for a part processing were randomly selected from the set of tool types.

Table 1. Parameters of Test Problems

Parameters	Scenario 1	Scenario 2
No. of tool types (ℓ)	20	30
No. of machining centers(i)	2	3
No. of parts(n)	40,60	60,90
Magazine capacity(s_i)	20	20
Sample size	20	20

Both the algorithm and the heuristic for the tool planning problem were coded in the FORTRAN language and the programs were run on a Vax 11/750 computer. Subproblem 1 of the proposed algorithm was solved by the LP subroutine ZX3LP in the IMS Library. The computation time for the algorithm and the heuristic is expressed by CPU time and is summarized in Table 2.

The proposed algorithm requires only marginally increased computational efforts compared to the heuristic rule. However, as problem size increases, computation time for the proposed algorithm increases dramatically.

The tool movement index (F) for the tool copy decision problem is used as a performance measure of the algorithm. F is the percentage ratio of tool movements resulting from the algorithm to tool movements due to the heuristic.

Table 2. Computation Time (Seconds)

Scenario	No. of parts	Algorithm	Heuristic
Scenario 1	40	30.3	0.29
	60	33.7	0.67
Scenario 2	60	325.4	0.79
	90	360.3	1.98

F is expressed as follows:

$$F = (Z_{alg} \,/\, Z_{heu}) * 100\% \,. \qquad (10)$$

Z_{alg} denotes the number of tool movements from the algorithm and Z_{heu} denotes the number of tool movements from the heuristic. A lower value of F represents better performance of the algorithm compared to the heuristic. The performance of the algorithm for each scenario is shown in Table 3.

The algorithm performed better than the heuristic at all duplication factor levels and part set size for all problems. In order to support the statement that the algorithm performs better than the heuristic, a standard t-test was carried out. Test results confirmed that the algorithm performs better than the heuristic at a 95% confidence level. As the γ value increases (more number of tool copies are allowed), the algorithm reduced the number of tool movements to a greater degree. The algorithm showed no difference in performance with different numbers of parts examined for all problems.

Table 3. Performance of the Algorithm

Scenario	γ	No. of parts	F(%)
Scenario 1	0.3	40	60.6
		60	60.2
	0.5	40	51.4
		60	47.5
	0.7	40	29.7
		60	34.0
Scenario 2	0.3	60	66.0
		90	67.2
	0.5	60	59.3
		90	60.5
	0.7	60	52.8
		90	53.7

6. Conclusions

In this chapter the tool planning problem, which determines the number of tool copies for each tool type when the total number of tool copies are limited, was concidered. Since the loading is dependent upon the tool set, the problem was mathematically formulated and solved in order to further contribute to the objective of reducing makespan. The mathematical formulation is a nonlinear integer programming problem which is partitioned into two subproblems. The constraint set of subproblem 1 was proved to be totally unimodular and subproblem 1 can be solved as a linear programming problem. Subproblem 2 is the same as that of the loading problem. Thus, the problem is solved by a similar procedure to the loading problem.

The performance of the algorithm is compared to that of a heuristic procedure for the randomly generated problems. The algorithm dominates the heuristic for all duplication factor levels and part volumes. The more tool copies allowed, the better the algorithm could reduce the number of tool movements.

In this paper, a method of determining the number of tool copies for each tool type is presented when the total number of tool copies is limited. The problem is formulated as an optimization problem with an objective of minimizing the number of tool movements while maintaining a given level of workload balance. The solution of the problem specifies the number of tool copies for each tool type and desired tool loading among machining centers. As a solution procedure, the problem is partitioned into two subproblems and solves them iteratively until convergence is observed.

References

1. Ammons, J. C., Lofgren, C. B., and McGinnis L. F., "A Large Scale Loading Problem in Flexible Assembly," Annals of O.R., 3, 319-328(1985).
2. Amoko-Gyampah, K., Meredith J. R., and Raturi, A., "A Comparsion of Tool Management Strategies and Part Selection Rules for a Flexible Manufacturing System," International Journal of Production Research, 30, 733-748(1992).
3. Berrada, M., and Stecke, K.E., "A Branch and Bound Approach for FMS Machine Loading," Proceedings of the First ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems : Operations Research Applications, University of Michigan, Ann Arbor, Aug.15-17, 156-171(1984).
4. Cook, N. H., Computer-Managed Parts Manufacture, Scientific American, 232, 22-29(1975).
5. Garey, M.R., and Johnson, D.S., Computers and Intractability : A Guide to the Theory of NP-Completeness, W. H. Freeman Press, New York(1979).
6. Han, M., Na, Y.K., and Hogg, G. L., "Tool Loading and Real Time Control in FMS," International Journal of Production Research, 27, 1257-1267(1989).
7. Horowitz, E. and, Sahni, S., Fundamentals of Computer Algorithms, Computer Science Press(1984).
8. Hwang, S., "A Constraint-Directed Method to Solve the Part Selection Problem in Flexible Manufacturing Systems," Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems : Operations Research Models and Applications, University of Michigan, Ann Arbor, Aug 12-15, 297-309(1986).
9. Papadimitriou, K. S., and Steiglitz, K., Combinatorial Optimization : Algorithms and Complexity, Prentice-Hall, New Jersey(1982).
10. Stecke, K. E., "Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems," Management Science, 29, 273-288(1983).
11. Tang, C.S., "A Job Scheduling Model for a Flexible Manufacturing Machine, 1986 IEEE International Conference on Robotics and Automation," 1, 152-155(1986).
12. Zavanella, L., and Bugini, A., "Planning Tool Requirements for Flexible Manufacturing : an Analytical Approach," International Journal of Production research, 30, 1401-14149