An Integrated Approach of Process Plan Selection and Scheduling
- 공정계획과 일정계획의 통합적 접근 -

Lee, Kyung Huy[*]
이 경 휘
Kim, Byoung Kug[*]
김 병 국

Abstract

본 연구에서는 공정계획과 일정계획을 순차적으로 수행하는 기존의 방법과는 달리, 공정계획의 선정과 일정계획의 동시적 접근을 통하여 생산시스템의 효율을 향상시키기 위한 통합방법론을 제시하고자 한다. 공정계획과 일정계획의 순차적 접근방법에서는 공정계획과 일정계획이 각각 지역적 판단기준(local criteria)에 의해 순차적으로 선정되어 생산 시스템의 전역적 판단기준(global criteria)와 상충(conflicts)되는 경우가 자주 발생하여 시스템의 폭주(congestion)나 재계획 등의 원인이 되고 있다. 이와같은 문제점들을 극복하기 위해 본 연구에서는 통합적 접근을 통해 전역적 판단기준에 의한 공정계획과 일정계획의 동시접근방법을 제시하고자 한다.

1. Introduction

In the traditional approach process planning and scheduling are performed in a sequential manner, in which an optimal process plan is generated by the technical criteria of process planning and then scheduling is thereafter done under the process plan constraints. A process plan generated cannot consider shop floor status about machine load, capacity, and failure. The sequential approach results in machine congestion or re planning due to missing shop floor status about machine load, capacity, and failure etc. in selecting the process plan (*Lenderink and Kals 1992*). Furthermore, scheduling is constrained on the domain of a process plan selected in process planning.
To overcome the problem of 'local optimalty', in this paper, an integrated approach to process plan selection and scheduling has been proposed, in which a process plan and a corresponding schedule are concurrently determined by global selection criteria considering the current shop floor status. Timed petri nets and token player algorithms are used to formally model the problem of 'process plan and schedule selection' and then select a process plan and schedule.

* *KyungPook Sanup University, Department of Industrial Engineering*

An illustrative example is followed for explaining the feasibility and validity of the proposed approach.

2. Integrated Model of Process Planning and Scheduling

An integrated process planning and scheduling model proposed here, as shown in Figure*1*, is describied as a concurrent process plan selection and scheduling based on process plan alternatives (*Lee 1996*). In the concurrent model, process planning generates an alternative process plan (in form of *AND/OR* Petri net) which includes machine and process flexibility and the process plan is then transformed into a corresponding dynamic process plan which models the 'process plan selection and scheduling' problem. A process plan and schedule is simultaneously selected on the dynamic process plan, with considering shop floor status, that is: machine load, capacility, and availability. The integrated model guarantees a decision making on process plan selection and scheduling which supervises a shop floor on line. For example, machine overloading or failure is incorporated in the decision making process.

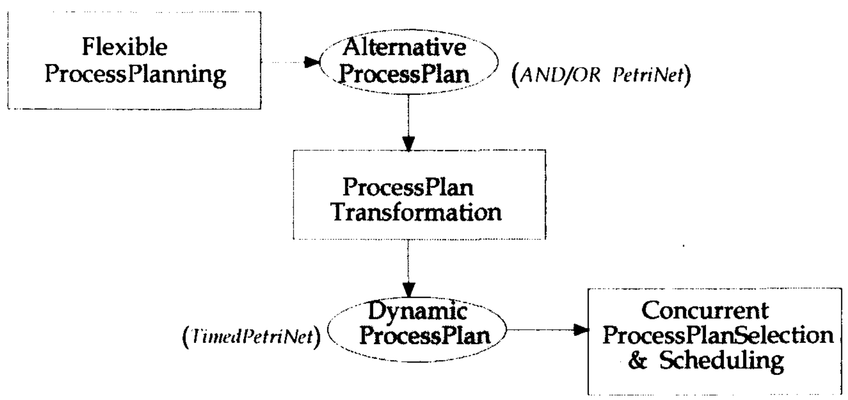

Figure*1*. Integrated model of process planning & scheduling

3. Process Plan Selection and Scheduling

As explained before, process plan selection and scheduling in the integrated model are concurrently executed on a Timed Petri net which represents a dynamic process plan. In this chapter, a Timed Petri net (*TPN*) model is briefly explained and a solution procedure on process plan selection and scheduling in the following.

3.1 *TPN* Model of Dynamic Process Plan

As explained before, an alternative process plan generated in process planning is transformed into a corresponding dynamic process plan, which models the following:

1) processes and their sequence,
2) machines and their status,
3) processing time for a specified [machine, process], and
4) machine/process status during time progress.

TPNs are a good tool for modeling a discrete event-based dynamic system (*DEDS*), specifically, the unique one for explicitly describing time-phased heuristic problems in that Petri nets are the mixed representation model od states and events (*Peterson 1981, Silva 1991*). Figure2 illustrates the *TPN* model of a dynamic process plan for producing a given part.

In the *TPN* strucure, shown in Figure2, places denote processes and machines, transitions represent process sequencing logic(*immediate transition*) and processing (*timed transition*), and arcs connecting places and transitions denote input or output function from places to transitions or *vice versa.* A token in a place denoting a process means a process state(*ready*), a token in a place representing a machine a corresponding machine state. Token distribution at time *t* shows the spectrum of system status at that time. In the *TPN*, process places are symbolized by small circles, machine places by small rectangles, immediate transitions by thin lines, and timed transitions by thick lines, as can be seen in Figure2. From the time phased marking progress of *TPN*, without ambiguity, we can explicitly deduce system status, job status, machines status, and process status such as the number of remaining parts, the remaining processes of parts in *WIP*(Work_In_Progress, machine load and availability, remaining time of machine/process in processing.

The *TPN* model of a dynamic process plan, therefore, guarantees a decision on process plan selection and scheduling which considers system status on line. It results in overcoming the overloading or replanning problem of a sequential approach.

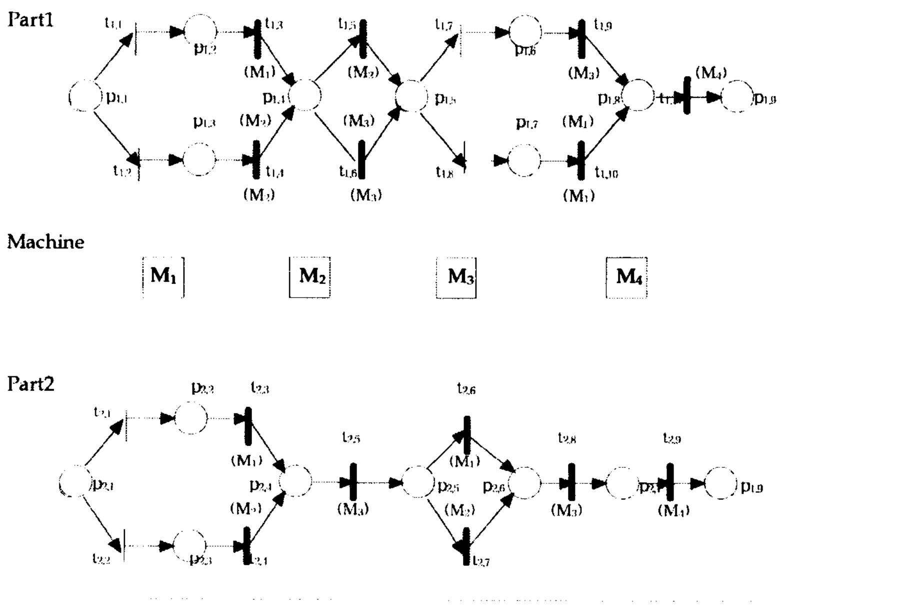

Figure2. *TPN* model of a dynamic process plan

3.2 Solution Procedure of Process Plan Selection and Scheduling

Notations:

t	*current time*
n_p	*the number of parts to be considered*
n_m	*the number of machines to be included*
$MS(t)$	*machine state vector at time t, for all machines* n_m
$PS(t)$	*job process state vector at time t, for all parts* n_p
$M(t)$	*marking vector at time t*
$FST(t)$	*firing status table at time t(FST means a table of fired, but uncompleted timed transitions and their remaining times)*
EIS_i	*a set of enabled, immediate transitions of part i*
CIS_i	*a set of enabled, immediate transitions in conflict, of part i*
ETS_i	*a set of enabled, timed transitions of part i*
CTS_i	*a set of enabled, timed transitions in conflict, of part i*

The basic solution procedure of process plan selection and scheduling is based upon the *token player algorithm* of *TPN*. In advancing the token player algorithm of *TPN*, conflicts (in case that the number of output transition of the place is greater than or equal to *2*) occur in various situations. Table*1* shows conflict types, problems, and detection rules. Conflicts are detected and then identified by rules listed in table1. In resolving conflicts, decisions are made in order to *1*) find the next job status (*process path selection problem*) and *2*) assign free machines to the processes of multiple parts which are selected (*concurrent assignment problem*). The *process path selection* problem is associated with execution of *enabled, immediate transitions in conflict* and the *concurrent machine assignment* problem *enabled, timed transitions in conflict*. Conflicts are classified into a *single conflict* or a *multiple conflict* which consist of *sequence conflicts(part/process sequencing)*, *interpart machine conflcits (machine selection)*, and *intrapart machine conflicts(machine allocation)*. Therefore, the problems identified are resolved by heuristics based on shop floor status.

Table*1*. Conflicts

conflict	transitions in conflict	problem	detection & identification
Single Conflict(SC)			$(\exists p) \mid O(p) \geq 2$
process conflict	immediate transition	process path selection	$(\exists p)$p.type= 'DUMMY' & $O(p) \in T_{immediate}$
sequence conflict	timed transition	part/process sequencing	$(\exists p)$p.type='CONTROL' & $O(p) \in T_{timed}$
interpart machine conflict	timed transition	machine selection	$(\exists p)$p.type='PROCESS' & $O(p) \in T_{timed}$
intrapart machine conflict	timed transition	machine allocation	$(\exists p)$p.type='MACHINE' & $O(p) \in T_{timed}$
Multiple Conflicts:			$(\exists t) t \in \{SC_1,\ldots,SC_n\}$, $n \geq 2$
machine/sequence conflict	timed transition	concurrent sequencing/ selection/allocation	

The conflict resolution is done by applying the corresponding heuristic for a conflict each. For example, the process path selection problem (or the machine allocation problem) may be resolved by the *Earliest Finishing Time (EFT)* of CIS_i (or CTS_i) and the machine selection problem by the *Min Process Time*. The concurrent assignment problem in multiple conflict is resolved by the two-phase heuristic which follows:

PhaseI) Problem Reduction:
In case for a multiple conflict to be identified, the concurrent machine assignment problem is first reduced as possible.

Problem Reduction Algorithm {

step 1) Construct the Process/Machine Incidence (*PMI*) matrix where its entry $a^i_{j,k}$ of row i and column (j,k) is described as

$$a^i_{j,k} := \begin{cases} 1, & \text{if the process } k \text{ of part } j \text{ requires machine } I \text{ which is currently available;} \\ 0, & \text{otherwise;} \end{cases}$$

step 2) Cross out the corresponding rows and columns of all entries satisfying one of the following cases:

C1) if the number of parts for machine M_i *is 1* (P_j)*, assign* M_i *to* P_j*;*
C2) if the number of machines for part P_j *is 1* (M_i)*, assign* M_i *to* P_j*;*

step 3) Select the next highest alternative machine of *EFT* for the processes (columns) of the *PMI* matrix, if it exists, and then replace it in the *PMI* matrix;

step 4) Apply step *2* to the modified *PMI* matrix until there exist no such cases.
}

The concurrent assignment problem is reduced into the machine selection problem, the machine allocation problem, or the mixed problem of the both (it means a non decomposible concurrent assignment problem).

Phase II) Dispatching:
Apply the corresponding heuritic for resolving the problem reduced. For example, the non decomposable concurrent assignment problem may be resolved by applying the EFT heuristic for all machine/process cases.

4. An Illustrative Example

An example, which consists of *'Part1 with face and slot features'* and *Part2 with face and pocket features'*, is given to illustrate the integrated process plan selection and scheduling, as already shown in Figure2. In the dynamic process plan, all operation times (timed transitions) are also given in Table2. Under the *TPN*, process plan selection and scheduling can be concurrently performed in compatible with the token player algorithm of the *TPN*. Initially, the first marking vector, *M(0)* is easily identified from the *MS(0)*. That is, machine places with marking denote currently free machines and all intial and first process places are marked with the same number of tokens as the production quantity of the part. From the marking vector, *M*, the next processes for free machines are determined from the enabled transitions and operation times. The production progress such as machine breakdowns or process ends is updated instantaneously and then explicitly reserved in the *TPN*.

Table2. Machine and time of process activity

part	process activity transition	machine	operation time
$Part_1$	$t_{1,3}$ (Face)	M_1	1
	$t_{1,4}$ (Face)	M_2	1
	$t_{1,5}$ (Rough_Slot)	M_2	1
	$t_{1,6}$ (Rough_Slot)	M_3	1
	$t_{1,9}$ (Semifinish_Slot)	M_3	3
	$t_{1,10}$(Semifinish_Slot)	M_1	3
	$t_{1,11}$(Finish_Slot)	M_4	3
$Part_2$	$t_{2,3}$ (Face)	M_1	1
	$t_{2,4}$ (Face)	M_2	1
	$t_{2,5}$ (Rough_Pocket)	M_3	3
	$t_{2,6}$ (Semifinish_Pocket)	M_1	1
	$t_{2,7}$ (Semifinish_Pocket)	M_2	1
	$t_{2,8}$ (Finish_Pocket)	M_3	3
	$t_{2,9}$ (Superfinish_Pocket)	M_4	3

Table3 shows the overall simulation of the example. An initial marking vector, *M(0)*, is described as follows:

$$M_0 = \begin{pmatrix} M_{part1}(0) \\ M_{part2}(0) \\ M_m(0) \end{pmatrix} = \begin{pmatrix} 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ 1\ 1\ 1\ 1 \end{pmatrix}$$

where $M_{parti}(0)$, $i = \{1,2\}$, is an initial submarking vector of process places of part *i* denoting initial job status, *PS(0)*, and an initial submarking of machine places, *MS(0)*. Under the current *PS(t)* & *MS(t)*, the first thing to be done is to resolve process path conflicts, that is: $\{t_{1,1}, t_{1,2}\}$ and $\{t_{2,1}, t_{2,2}\}$. In both cases, ties occur and then it tries to break the tie of the first conflict set $(t_{1,1})$. Thereafter, $t_{2,2}$ is selected from the *EFT* heuristic. After all *CISs* have been resolved and fired, *ETS* are $\{t_{1,3}, t_{2,4}\}$. Both transitions start firing and then put into the *FST*. Since there exist no *ETS*, time advances to the earliest time of *FST(1)* and both transitions are completely fired. When $t = 1$, a concurrent assignment problem which is a mix of an interpart machine conflict and an intrapart machine conflict occurs. By applying the concurrent assignment resolution algorithm, $t_{2,5}$ (M_3) and $t_{1,5}$ (M_2) are selected. This procedure proceeds successively until enalbed transitions $(EIS \bigcup ETS)$ & *FST* become empty. The process plan and schedule selected is listed in Table*3*. The total makespan of the process plan and schedule is *10* and their operation time *19*.

Table*3*. Simulation

No.	time	transition			Machine	Conflict type and problem
		enabled	in_conflict	fire		
1	0	$t_{1,1}$, $t_{1,2}$	$\{t_{1,1}, t_{1,2}\}$	$t_{1,1}$	M_1	process path selection
		$t_{2,1}$, $t_{2,2}$	$\{t_{2,1}, t_{2,2}\}$	$t_{2,2}$	M_2	process path selection
2		$t_{1,3}$, $t_{2,4}$	-	$t_{1,3}$, $t_{2,4}$	M_1,M_2	-
3	1	$t_{2,5}$	-	$t_{2,5}$	M_3	-
		$t_{1,5}$, $t_{1,6}$	$\{t_{1,5}, t_{1,6}\}$	$t_{1,5}$	M_2	concurrent assignment
4	2	$t_{1,7}$, $t_{1,8}$	$\{t_{1,7}, t_{1,8}\}$	$t_{1,8}$	M_1	process path selection
5		$t_{2,6}$, $t_{2,7}$	$\{t_{2,6}, t_{2,7}\}$	$t_{2,7}$	M_2	machine selection
6	4	$t_{1,11}$	-	$t_{1,11}$	M_4	-
		$t_{2,8}$	-	$t_{2,8}$	M_3	-
7	7	$t_{2,9}$	-	$t_{2,9}$	M_4	-
8	10	Ø	-	-	-	-

* TOTAL MAKESPAN = 10
**TOTAL OPERATION TIME = 19

From the illustrative example, it seems that the integrated process plan selection and scheduling based on *TPN* is a promising tool for integrating process planning and scheduling.

5. Concluding Remarks

An integrated approach of process plan selection and scheduling has been proposed in this paper. In the integrated approach, process plan selection and scheduling were concurrently executed on-line with considering shop floor status, that is: machine load, capacity, or availability. Timed Petri nets were used to model a dynamic process plan based on process plan alternatives. A process plan and a schedule were simultaneously selected from the dynamic process plan.

It was shown that the concurrent process planning and scheduling approach proposed here resolved the 'local optimality' problem of the sequential approach that results in overloading or replanning.

References

1. N.E.Larson and L.Alting,"Dynamic Planning Enriches Concurrent Process and Production Planning," *International Journal of Prduction Research*, Vol.30, No.8 (1992)

2. K.H.Lee,"Formal Integration of Process Planning and Shop Floor Control for CIM," *Proceedings of the KIIE Conference* (1996)

3. Lenderink and Kals, "The Integration of Process Planning and Machine Loading in Small Batch Part Manufacturing,"*Robotics and Computer Integrated Manufacutring*, Vol.10, No.1 2 (1993)

4. J.L.Peterson, *Petri net theory and the modeling of systems*, Prentice-Hall, NewJersey (1981)

5. M.Silva,"Petri Nets and Flexible Manufacturing," *Concurrency and Nets* (1991).

6. H.K.Tonshoff,U.Beckendorff and N.Anders,"FLEXPLAN-A Concept for Intelligent Process Planning and Scheduling," *CIRP International Workshop on Computer Aided Process Planning* (1989)

7. H.Zhang,"IPPM-A Prototype to Integrate Process Planning and Jobshop Scheduling Functions,"*Annals of CIRP*, Vol.42, No.1 (1993)