

UNFOLDING을 이용한 유연생산시스템의 JOB-SHOP스케줄링 분석

김정원^o, 이종근
창원대학교 전자계산학과 정보통신연구실
경남 창원시 사림동 9, 641-773
Tel, Fax : +82-551-79-7420
e-mail: jklee@sarim.changwon.ac.kr

JOB-SHOP SCHEDULING ANALYSIS IN FLEXIBLE MANUFACTURING SYSTEM USING UNFOLDING

Jung-Won Kim, Jong-Kun Lee
LICS / Department of Computer Science Changwon National University,
Changwon, Kyungnam, 641-773, South Korea

요약

본 연구는 TPN unfolding을 이용하여 WIP의 FMS(Flexible Manufacturing System)를 분석하는 방법을 제시한다. PN의 unfolding은 상태폭발이 발생하지 않는 concurrent system의 검증에 사용되는 순서기반방법이다. 본 연구는 일반적으로 발생하는 순환상태스케줄문제에서 가장 그 작업과정 시간을 최적화함을 위하여 원래의 net을 동일한 비순환 net으로 바꾸어 줄 수 있는 unfolding 개념을 기반으로 한 것이다.

1. 서론

단일 machine 또는 여러 machine들의 서브프로세스망으로 구성되는 FMS에 있어서 가장 중점적으로 다루어지는 것이 어떻게 효과적으로, 최적으로 스케줄을 할 수 있는가 하는 것이다. 뿐만아니라 단순한 시스템 형태에 있어서도 이러한 스케줄문제는 복잡성을 띄기때문에 여러가지의 스케줄방법 [1,3,7,10,11]이 여러 연구가들에 의해 연구되어져 왔다. 우리는 WIP(Work In Process)의 최소화와 최적순환시간의 결정을 위한 순환스케줄분석에 초점을 맞추었다. unfolding은 Petri Nets을 unfolding 함으로 얻어지는 것이고 unfolding된 Petri Nets은

원net의 도달가능정보를 가지고 있다. unfolding된 net상에서의 구조적분석은 원 net상에서 분석하는 것보다 훨씬 더 쉽다. unfolding은 부분순서의 미에 기반을 두기 때문에 상태공간폭발을 피할 수 있는 것이 장점이다.

본 연구는 FMS의 순환스케줄문제를 해결하기 위하여 Time Petri Nets의 unfolding을 기반으로한 효과적인 방법을 제시한다.

2장에서는 Time Petri Nets의 정의와 부분적인 unfolding에 대하여 소개하고 3장에서는 FMS의 모델을 소개하고 페쇄루프모델의 서로다른 방법등을 강조할 뿐만아니라 초기모델에서 unfolding된

time Petri Nets 모델로 변환되는 동안에 발생하는 스케줄문제를 다루게 된다. 그리고 4장의 결론으로 이루어 진다.

2. Time Petri Nets

Time Petri Nets의 이론은 널리 연구되어져 왔고 본 연구에서는 기본적인 정의[2.4,5]만을 다룰 예정이다.

2.1 Time Petri Nets(TPN))의 일반적 정의

TPN은 (P, T, E, S, M_0, τ) 로 구성되는데 여기에서, $P = \{p_1, p_2, \dots, p_n\}$, $|P| \neq 0$, P는플레이스의유한집합, $T = \{t_1, t_2, \dots, t_m\}$, $|T| \neq 0$, T는트랜지션의유한집합, $P \cap T = \emptyset$, $E : P \times T \rightarrow N$, E는 입력함수, $S : T \times P \rightarrow N$, S는 출력함수 (N :양의 정수집합), $M_0 \in M = \{M \mid M : P \rightarrow N\}$, M_0 는 초기토큰상태, τ 는시간함수: $\tau : T \rightarrow Q^* \times (Q^* \cup \infty)$ (Q^* 는양의정수집합)

만일 TPN에 cycle이 존재하지 않는다면 TPN은 acycle하다 한다. $S = \langle M, I \rangle$ 를 TPN의 현재상태라 가정하면 트랜지션간의 점화순서를 다음의 세 가지 형태로 정리할 수 있다.

1) 트랜지션 $t_e \in T$ 는 다음의 식을 만족할 경우 트랜지션 $t_s \in T$ 와 순차관계이다. 여기서 p는 공통 플레이스의 집합이다.

$$C_s(t_e/t_s) = \{t_e, t_s \mid t_e \neq t_s, S(t_e) \cap E(t_s) \neq \emptyset = p \in P\}$$

2) 트랜지션 $t_e \in T$ 는 다음의 식을 만족할 경우 트랜지션 $t_s \in T$ 과 경합관계이다. 여기서 t_k 는 트랜지션 t_s 에서 나머지 점화가능 트랜지션들의 집합이다.

$$C_r(t_e/t_s) = \{t_s \mid t_s \neq t_e, (E(t_s) \cap E(t_k) \subseteq S(t_e) \neq \emptyset = p) \wedge E(t_s) \subseteq E(t_k) \wedge E(t_k) \subseteq E(t_e)\}$$

3) 트랜지션 $t_e \in T$ 는 다음의 식을 만족할 경우 트랜지션 $t_s \in T$ 와 병행관계이다.

$$C_e(t_e/t_k) = \{t_e, t_k \mid t_e \neq t_s, (E(t_s) \cap E(t_k) = \emptyset \wedge ((E(t_s) \cap E(t_k) \subseteq S(t_e)) = p)\}$$

2.2 Unfolding

Unfolding이라는 개념은 원래 McMillan[9,12]에

의해 제안된 것으로서 상태공간폭발이 발생하지 않는 concurrent system의 검증에 대한 "부분순서기반방법"에서 제기되었다[6,8].

정의 2.1(OCN:Occurrence net):모든 플레이스 $p \in P$ 에서 최소한개의 트랜지션을 갖을 때 $|p| \leq 1$, OCN $TPN = (P, T, E, S, M_0)$ 은 acycle net이 된다.

정의 2.2(Configuration):트랜지션의 집합 $C \subseteq T$ 은 OCN의 configuration라한다: 만일(1) $t' \in C$ 에서 C 가 모든 트랜지션 t' 의 선행흐름을 포함한다,(2) C 가 상호관계적 병합트랜지션을 포함하지 않는다.

정의 2.3: C 은 OCN의 configuration이다. C 의 마지막 마킹 $FM(C)$ 은 C 의 트랜지션을 점화하고 초기마킹에서부터 마지막마킹에 도달함을 나타낸다.

정의2.4:일반 패트리 네트는 유한한 마킹흐름을 갖는OCN인 unfolding으로 변형된다.

2.3 net의 성능평가

전체수행완료시간이 보통 스케줄이론에 있어 정상적인 성능평가가 되기 때문에 최적의 순서를 계산하는 것이 충분한 평가가 된다.

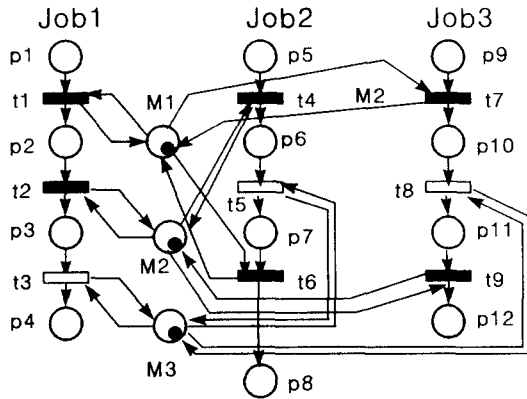
관련된 최적의 스케줄은 계산되어진 최적화 과정을 통하여 가능한 점화트랜지션에 의해 얻어진다. 이때 순차, 경합, 병행과 같은 관계형태를 기반으로 하는 도달가능시간을 고려할 수 있다. 순차관계의 경우는 아주 단순하기 때문에 순서관계를 비교할 필요가 없다. 경합또는 병행관계의 경우에는 순서관계에 있어서 기대되는 최대시간을 비교한 후 최소기대시간을 선택해야 한다.

3. FMS의 모델화

3.1 Job-Shop Scheduling modelling

하나의 시스템은 3개의 machine 즉, M1, M2, M3 그리고 3개의 job 즉, JOB1, JOB2, JOB3로 구성되는것으로 고려하고 이러한 job의 수행순서는 아래와 같다.

Job1 : (M1,5) (M2,6) (M3,1)
 Job2 : (M2,8) (M3,2) (M1,7)
 Job3 : (M1,4) (M3,3) (M2,5)



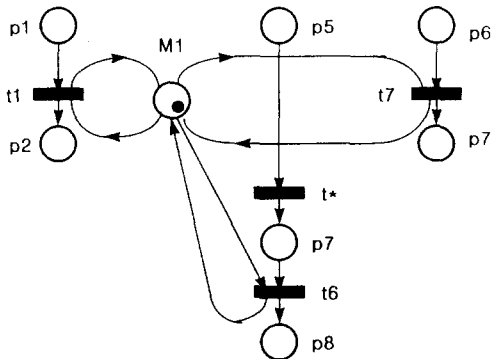
(그림 1) production model

(표 1) (그림 1)의 트랜지션표

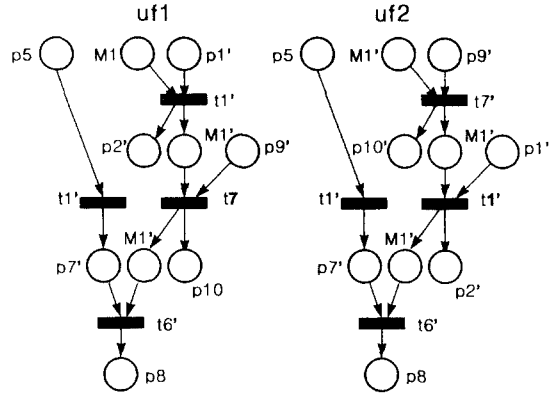
transition	time	works	relation
t1	5	Job1	Cc
t2	6		Cc
t3	1		Cc
t4	8	Job2	Cc
t5	2		Cc
t6	7		Cc
t7	4	Job3	Cc
t8	3		Cc
t9	5		Cc

1) machine 1(M1):

machine 1은 (그림 2)와 같이 2개의 처리 Job에서 3개의 작업을 포함한다. 그리고 Machine 3의 수행 unfolding net은 (그림 3)과 같이 2가지의 스케줄에 의해 설명된다.



(그림 2) Operation of M1



(그림 3) Unfolding nets of Operation M1

이러한 경우에 아래와 같은 2가지의 process를 보일 수 있다.

$$\text{Suf1: } t1 \rightarrow (t7 // t1^*) \rightarrow t6,$$

$$\text{Suf2: } t7 \rightarrow (t1 // t1^*) \rightarrow t6,$$

여기서 t^* 은 포함된 몇가지의 process를 의미한다.

이 수행의 반환시간은 다음과 같다.

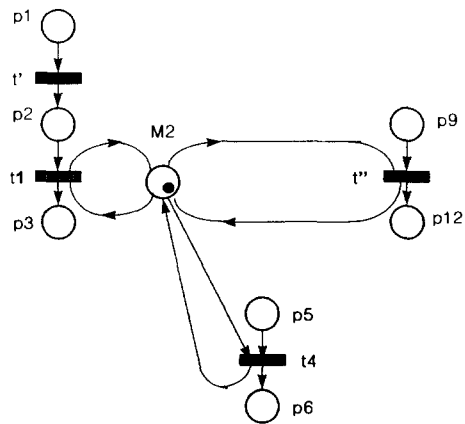
$$\text{Tuf1} = 5 + \max(4 // 10) + 7 = 22$$

$$\text{Tuf2} = 4 + \max(5 // 10) + 7 = 21$$

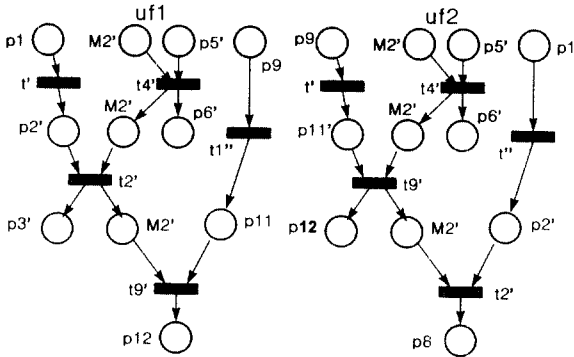
여기서, machine 1에서 발생하는 2가지의 수행에서 process uf2가 uf1보다 좀 더 유용하다는 것을 알 수 있다.

2) machine 2(M2):

machine 2는 (그림 4)와 같이 3개의 처리 Job에서 3개의 작업을 포함한다. 그리고 Machine 2의 수행 unfolding net은 (그림 4)과 같이 2가지의 스케줄에 의해 설명된다.



(그림 4) Operation of M2



(그림 5) Unfolding net of Operation M2

M2는 아래와 같은 2가지의 process를 보일 수 있다.

$$\text{Suf1:}(t1'/t4') \rightarrow (t2'/t11') \rightarrow t9'$$

$$\text{Suf2:}(t1'/t4') \rightarrow (t9'/t11') \rightarrow t2'$$

이 수행의 반환시간은 다음과 같다.

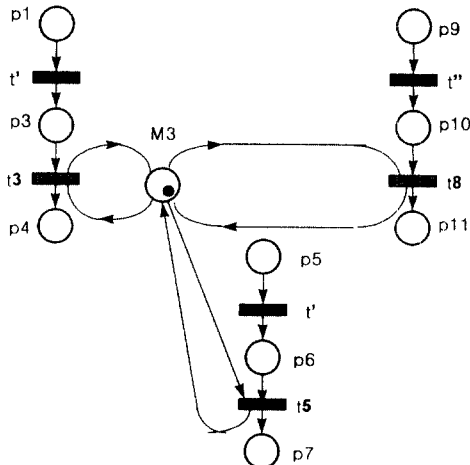
$$\text{Tuf1} = \max(5//8) + \max(6//11) + 5 = 24$$

$$\text{Tuf2} = \max(5//8) + \max(5//11) + 6 = 25$$

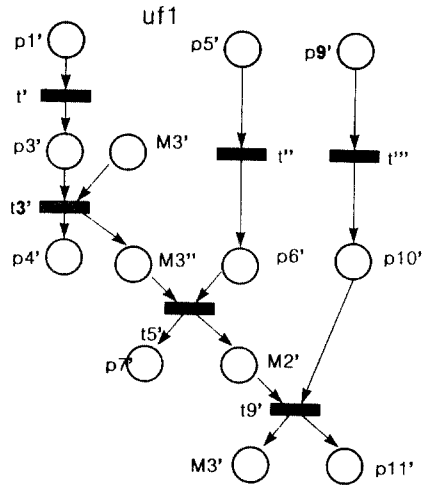
이것으로 볼 때 machine 2에서 발생하는 2가지의 수행에서 process uf1가 uf2보다 좀 더 유용하다는 것을 알 수 있다.

3) machine 3 (M3):

machine 3는 (그림 6)과 같이 3개의 처리 Job에서 3개의 작업을 포함한다. 그리고 Machine 3의 수행 unfolding net은 (그림 7)과 같이 2가지의 스케줄에 의해 설명된다.



(그림 6) Operation of M3



(그림 7) Unfolding net of Operation M3

Machine 3의 경우에는 6개의 프로세스가 존재하지만 이러한 프로세서들의 unfolding net의 형태들은 아래와 같이 모두 동일하다.

$$\text{Suf1:}t1' \rightarrow (t3//t'') \rightarrow (t5//t''') \rightarrow t8,$$

$$\text{Suf2:}t1' \rightarrow (t3//t''') \rightarrow (t8//t'') \rightarrow t5,$$

$$\text{Suf3:}t1'' \rightarrow (t5//t') \rightarrow (t3//t''') \rightarrow t8,$$

$$\text{Suf4:}t1'' \rightarrow (t5//t''') \rightarrow (t8//t') \rightarrow t3,$$

$$\text{Suf5:}t1''' \rightarrow (t8//t') \rightarrow (t5//t'') \rightarrow t3,$$

$$\text{Suf6:}t1''' \rightarrow (t8//t'') \rightarrow (t3//t') \rightarrow t5,$$

이 수행의 반환시간은 다음과 같다.:

$$\text{Tuf1} = 11 + \max(1//8) + \max(2//4) + 3 = 26$$

$$\text{Tuf2} = 11 + \max(1//4) + \max(3//8) + 2 = 25$$

$$\text{Tuf3} = 8 + \max(2//11) + \max(1//4) + 3 = 26$$

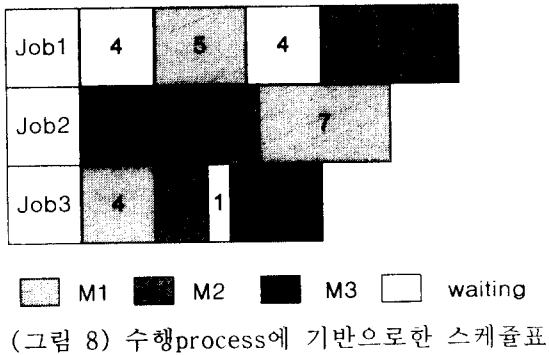
$$\text{Tuf4} = 8 + \max(2//4) + \max(3//11) + 1 = 25$$

$$\text{Tuf5} = 4 + \max(3//11) + \max(2//8) + 1 = 24$$

$$\text{Tuf6} = 4 + \max(1//4) + \max(1//11) + 2 = 30$$

여기에서 프로세스 uf5가 다른 프로세스 보다 좀 더 유용하다는 것을 알 수 있다.

결론적으로 (그림 8)과 같이 machine process작업에 기반으론 process 스케줄을 생성하였고 process에서 전체대기시간은 4+4+1=9이다.



4. 결론

본 연구는 Time Petri Nets unfolding을 이용한 FMS의 process분석에 관련한 방법을 제시하였다.

본 연구에서는 참고문헌[17]에서 제시한 production ratio를 고려하지 않았으며 unfolding 개념을 가지고 처음으로 스케줄링문제에 적용시켜 보았다. 이러한 unfolding개념의 적용은 genetic algorithm의 적용방법과 유사한 점을 발견 할 수가 있었는데, 앞으로 이러한 주제에 초점을 맞출 것이다.

참고문헌

[1] Berthomier B., Diaz M. (1991), Modeling and Verification of Time Dependent Systems using Time Petri nets, IEEE Tr. Soft Eng., 17, 259-273.

[2] Carlier J., Chrentienne P. (1985). Timed Petri nets Schedules, TR, Univ. P. et M..

[3] Holliday M., Vernon M. (1987). A Generalized TPN Model for Performance Analysis, IEEE. Tran. On s/w, se-13, 1297-1310.

[4] Lee J. (1997). Time Composition Problem in Time Petri Nets, In: Proceeding CIS'97, 1997, pp. 507-512.

[5] Valero V., Frutos D., Cuartero F. (1995). Timed Processed of Timed Petri Nets, In: Proceeding ATPN 1995, 490-509

[6] Zuberek W., Kubiak W. (1993). Thought Analysis of Manufacturing Cells Using Timed Petri nets, In: Proceeding ICSYMC 1993, 1328-1333

[7] Zuberek W. (1995). Schedules of Flexible Manufacturing Cells and their Timed Colored Petri net Models, In: Proceeding ICSYMC 1995, 2142-2147

[8] Kondratyev A., Kishinevsky M., Taubin A. and Ten s., "Analysis of Petri nets by Ordering Relations in Reduced Unfolding," In: Kluwer Aca. Pub., Boston, 1996

[9] Ohl H., Camus H., Castelain E. and Gentina JC, "Petri nets Modelling of Ratio-driven FMS and Implication on the WIP for Cyclic Schedules, In: Proceeding SMC'95, 1995, pp. 3081-3086

[10] Hwang CH, Lee DI, "A Concurrency Characteristic in Petri net Unfolding," In: Proceeding SMC'97, 1997, pp. 4266-4273

[11] Lee DY, DiCesare F. (1995) Petri Net-based heuristic Scheduling for Flexible manufacturing, In: Petri Nets in Flexible and Agile Automation, ed. Zhou MC, Kluwer Aca, pub. USA 149-187

[12] McMillan KL, "A technique of state space search based on unfolding," In: Formal Methods in System Design 6(1), 1995, pp. 45-65

[13] Korbaa O, Camus H. and Gentina JC, "FMS Cyclic scheduling with overlapping production cycles, 1997, pp. 35-52

[14] Richard P., Xie X., "Scheduling sequential and Flexible machines Using Timed Petri nets," In: Proceeding ICATPN'97, 1997, pp. 151-165

[15] McMillan. K, "A Technique of State Space Search Based on Unfolding," Formal Methods in System Design 6(1), 1995, pp. 45-65