

타임 패트리넷 Unfolding을 이용한 FMS의 스케줄링 분석

A Scheduling Analysis of FMS Using TPM Unfolding

이 종 근
(Jong Kun Lee)

Abstracts : In this paper, we are focused on the analysis of a cyclic schedule for the determination of the optimal cycle time and the minimization of work in process. Especially, this paper deals with the product ratio-driven cyclic scheduling problem in FMS with different products and ratio using the TPN(Time Petri Nets) unfolding.

Keywords : acycle, FMS, scheduling problem, slice, TPN, unfolding

I. 서론

유연생산시스템(Flexible Manufacturing System, FMS)이란 다품종 소량 생산을 그 특징으로 하는 제조 시스템이다. 이와 같은 생산 시스템의 구성은 대개 각종 CNC machine, 치공구, 공정간의 이동을 위한 장비들의 종합적인 net으로 되어 있다. 이와 같은 유연생산 시스템에 있어서 가장 중점적으로 다루는 것은 작업효율의 최적화를 위해 시스템을 구성하는 기기들의 작업 스케줄을 최적화하는 것이다. 이러한 시스템에 대한 스케줄을 모델링하고 분석하는 연구가 많은 연구가들에 의해 행해져 왔다[1] [5].

일반적으로 Time Petri Net(TPN)으로 모델링 한 FMS 모델의 스케줄 분석은 보통 cycle이 포함되어 있다. 특히 모델에 포함된 병행성을 다룰 수 있는 방법이 제공되지 않아 복잡한 병행적 모델의 도달성 분석 및 공유자원(shared resource)의 선점 등으로 인해 충돌이 발생하는 경우가 빈발하여 분석하는데 어려운 단점이 있다. 이러한 문제점을 극복하기 위하여 모델에 내재된 구조적 병행성을 상태불변(S-invariant) 개념을 적용 분석하여 여러 개의 slice들로 분할한다. 여러 개의 slice를 생성한 후 합성적 분석 기법을 이용하여 시스템의 교착상태, 플레이스의 유한성, 트랜지션의 수행가능성등을 검사한다. 또한 cycle이 포함된 원래의 TPN 모델을 acycle한 TPN으로 변형하는 unfolding 개념을 적용하여 분석함으로써 복잡성을 극소화 시킬 수가 있다. unfolding된 TPN의 분석은 부분순서의미에 기반을 두어 상태공간폭발을 피할 수 있어 원래의 TPN을 분석하는 것보다 훨씬 용이하다.

본 연구에서는 처음으로 unfolding 개념을 이용하여 유연생산시스템에서의 스케줄링 문제를 모델링하고 분석하고자 한다. 생산 비율을 달리하는 production ratio-driven FMS를 TPN을 이용하여 모델링 한 후, 이를 petri net slice개념을 적용하여 원래 net를 분할하고, 분할된 각각의 모듈을 다시 unfolding을 이용하여

동일한 acycle한 net으로 변형하여 스케줄을 분석한다.

본 연구의 구성은 2절에서는 TPN의 기본적인 특성과 TPN의 정의 및 unfolding에 대하여 설명하며, 3절에서는 petri net slice에 대해 설명하며, 4절에서는 FMS의 모델을 소개하며 특히 생산율을 달리하는 폐쇄 루프모델의 서로 다른 방법을 강조할 뿐만 아니라 초기 모델에서 unfolding된 TPN 모델로 변환하는 동안에 발생하는 스케줄 문제를 다루게 된다. 마지막으로 결론 및 앞으로의 연구과제가 5절에 주어진다.

II. TPN의 정의 및 unfolding

본 연구에서는 TPN의 기본적인 특성을 이미 알고 있는 것으로 가정하며 기본적인 정의[6] - [8]만 언급한다.

1. TPN의 일반적 정의

TPN은 $\langle P, T, E, S, Mo, \tau \rangle$ 인 6가지 튜플로 구성되며, $P = \{p_1, p_2, \dots, p_n\}$, $|P| \neq 0$, P는 플레이스의 유한집합, $T = \{t_1, t_2, \dots, t_m\}$, $|T| \neq 0$, T는 트랜지션의 유한집합, $P \cap T = \emptyset$, $E : P \times T \rightarrow N$, E는 입력함수, $S : T \times P \rightarrow N$, S는 출력함수(N:양의 정수집합), $Mo \in M = \{M | M : P \rightarrow N\}$, Mo는 초기토큰상태, τ 는 시간함수: $\tau : T \rightarrow Q^* \times (Q^* \cup \infty)$ (Q*는 양의정수집합)

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.

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

2) 트랜지션 $T_e \in T$ 는 다음의 식을 만족할 경우 트랜지션 $t_s \in T$ 와 경합관계이다. 여기서 tk는 트랜지션 t_s 에서 나머지 점화가능 트랜지션들의 집합이다.

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

3) 트랜지션 $t_e \in T$ 는 다음의 식을 만족할 경우 트랜

지선 $ts \in T$ 와 병행관계이다.

$$C_c(te/tk) = \{te, tk \mid te \neq ts, (E(ts) \cap E(tk) = \emptyset) \wedge ((E(ts) \cap E(tk) \subseteq S(te)) = p)\}$$

2. Unfolding

Unfolding이라는 개념은 원래 McMillan[9][10]에 의해 제안된 것으로 상태공간폭발이 발생하지 않는 concurrent system의 검증에 대한 “부분순서기반방법”에서 제기되었다[11][12].

정의 1(OCN: Occurrence net) : 모든 플레이스 $p \in P$ 에서 최소한개의 트랜지션을 가질 때 $|*p| \leq 1, OCN TPN = (P, T, E, S, Mo)$ 은 a cycle net이 된다.

정의 2(Configuration) : 트랜지션의 집합 $C' \subseteq T'$ 은 OCN의 configuration이라 한다 : 만일 (1) $t' \in C'$ 에서 C' 가 모든 트랜지션 t' 의 선행 흐름을 포함한다. (2) C' 가 상호 관계적 병합 트랜지션을 포함하지 않는다.

정의 3 : C' 은 OCN의 configuration이다. C' 의 마지막 마킹 $FM(C')$ 은 C' 의 트랜지션을 점화하고 초기마킹에서부터 마지막 마킹에 도달함을 나타낸다.

정의 4 : 일반 패트리 넷은 유한한 마킹흐름을 갖는 OCN인 unfolding으로 변형된다.

다음은 unfolding의 예이다. 그림 1의 (1)은 패트리 넷이고 (2)는 패트리 넷 (1)의 unfolding이다.

즉 패트리 넷의 마킹에서 사이클을 배제하여 표시하고 마킹흐름에서 같은 마킹의 반복되어지는 부분을 찾아서 유한한 마킹으로 변형한 넷이 바로 unfolding이다.

따라서, unfolding으로 변형된 넷은 순차형의 마킹 흐름을 가지므로 이를 가지고 넷의 성질을 분석하기가 쉬어진다.

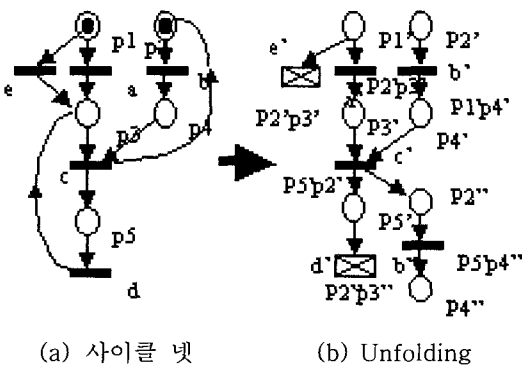


그림 1. 패트리 넷과 unfolding.
Fig. 1. Petri net and unfolding.

III. Petri net slice

모델분할은 극소 상태불변을 기본 단위로 하여 시스템을 나누고 기본 단위에 포함되지 않는 플레이스들의 정보는 인근 극소 상태불변에 추가하여 모든 플레이스들이 기본단위에 나타나도록 한다[14].

정의 5(상태불변(S-invariant)) : N을 P/T net이라

할 때, $N' \bullet i = 0$ 를 만족하는 플레이스 벡터 $i : S_N \rightarrow Z$ 를 상태불변이라 한다.

N' 은 패트리 넷 모델을 행렬(트랜지션 X 플레이스)로 나타낸 것이다.

정의 6 극소 상태불변(minimal invariant) : 상태불변으로 양의 값으로만 표현되며 다른 상태불변을 포함하지 않는 극소 형태(minimal)를 말한다.

정의 7 (Petri net Slices) : 전체 모델 $N = (P, T, F, W, M)$ 에서 모델 분할 알고리즘에 의해 분할된 플레이스 집합을 SliceSet = $\{P_Slice_i \mid i = 1 \dots n\}$ 라 하면, Petri net Slices은 $\{Slice_i \mid i = 1 \dots n\}$ 로 정의되고 각각의 Slice는 Petri net Slice_i = $(P_i, T_i, F_i, L_i, W_i, M_i)$ 는 아래 조건을 만족한다.

$$P_i = P_Slice_i, \\ L_i : T_i \rightarrow \Sigma^+ \text{는 트랜지션에 문자열의 레벨을 정의한 함수,} \\ T_i = \{t \in T \mid s \in P_i, (s, t) \in F \text{ 혹은 } (t, s) \in F\}, \\ F_i = \{(p, t) \in F, (t, p) \in F \mid p \in P_i, t \in T_i\}, \\ \forall p \in P_i, W_i(p) = W(p) \text{이고 } \forall p \in M_i, M_i(p) = M(p).$$

Petri net 모델에서 트랜지션 t 의 수행조건은 t 에 연결된 이전 플레이스들($*t$)에 토큰이 존재하여야 한다. Petri net slices는 여러 subnet으로 분할되어 있으며 트랜지션의 레벨에 의해 subnet 간의 동기화가 이루어진다. Petri net slices에서 트랜지션의 수행조건은 아래와 같이 정의할 수 있다.

$L_i(t)$ 가 공유되지 않을 때 : 트랜지션 t 의 수행조건만 만족하면 된다.

$L_i(t)$ 가 여러 Slice에 공유될 때 : $L_i(t)$ 를 가지는 모든 트랜지션들이 수행조건을 만족하여야 한다.

IV. FMS 모델링

1. Ratio-driven FMS modeling

하나의 시스템은 M1, M2, M3, M4라는 네 개의 machine으로 구성되어 있고 OP1, OP2, OP3이라는 세 개의 작업이 수행되며, 각각의 작업 생산율을 1:1:3이라 가정한다. 특히 OP1의 경우 3개의 공정에 M3, M1, M4의 기기가 필요하며, OP2의 경우 4개의 공정에 기기 M3, M2, M4가 필요하다. OP3의 경우에는 M2와

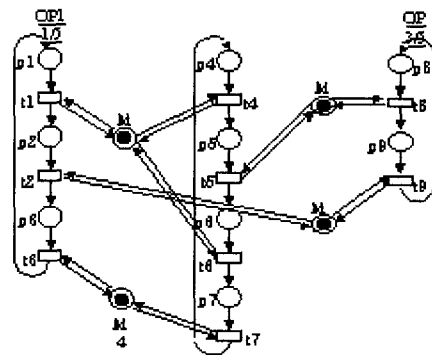


그림 2. 서로 다른 생산율을 갖는 FMS model.
Fig. 2. Ratio-driven FMS model.

M1의 기기만 필요하다.시스템은 그림 2와 같이 모델링되며 각 공정의 작업시간은 표 1로 표시된다.

작업생산율이란 세 개의 작업 활동에서 각 작업활동이 갖는 가중치라고 생각하면 좋겠다. 즉 OP1과 OP2가 1회의 작업을 하는동안 OP3은 3회의 작업이 가능한 것을 의미한다. 따라서, 전체적인 확률을 본다면 OP1은 1/5, Op2는 1/5 그리고 Op3은 3/5의 작업 호라동이 이루어지는 시스템이다. 표 1에는 각 트랜지션의 성질을 표시하였는데, 각 트랜지션에서의 소요시간과 관련 작업 그리고 각 트랜지션의 점화형태를 표시하였으며 관련되어진 트랜지션도 같이 나타내었다.

표 1. 그림 2의 트랜지션표.
Table 1. Transition table of Fig. 1.

트랜지션	시간	작업	관계	트랜지션
t1	13	OP1	Cc	t4
t2	10		Cc	t9
t3	11		Cc	t7
t4	20	OP2	Cc	t1
t5	7		Cc	t8
t6	6		Cc	t1,t4
t7	14		Cc	t3
t8	4	OP3	Cc	t5
t9	7		Cc	t2

2. 생산 비율을 병렬로 구성한 모델링

그림 1에서 생산율을 달리하는 작업 OP3을 병렬로 나열하여 모델링하면 그림 2와 같다[8].

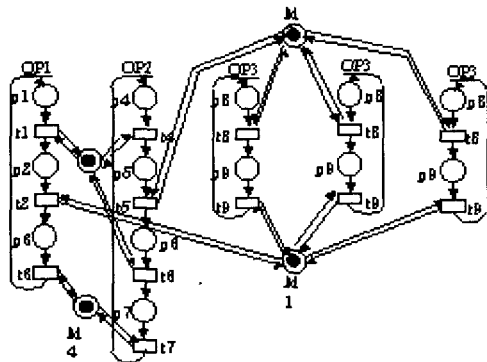


그림 3. 병렬로 표현한 모델.
Fig. 3. Parallel model of OP3.

즉 Op3가 3회의 작업활동을 하게 되므로 이를 각각의 별도의 작업활동으로 간주한다면 그림 3과 같은 5개의 작업활동 형태로 나타 낼 수가 있다.

이제 그림 3의 모델에서 기기별로 부-패트리 넷으로 구분하여 각 기기모델에서의 unfolding넷을 표시하여 분석한다.

1) M3에 대한 모델링 및 unfolding nets : M3은 그림 4와 같이 OP1과 OP2의 두 개의 작업에서 트랜지

션 t1, t4, t6 세 개의 공정에 이용된다. 그림 4는 M3에 대한 unfolding nets를 나타내고 있다.

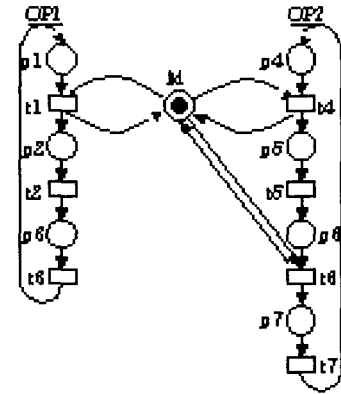


그림 4. M3 동작.
Fig. 4. Operation of M3.

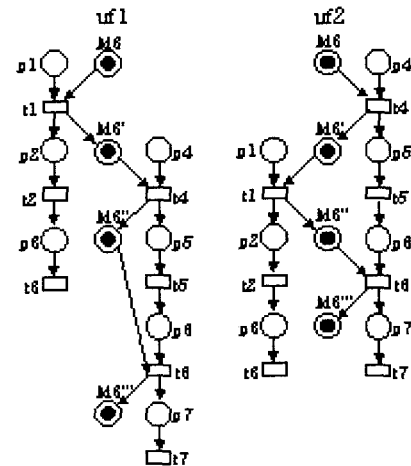


그림 5. M3에 대한 unfolding nets.
Fig. 5. Unfolding net of M3.

그림 5에서와 같이 트랜지션 t6는 t4가 수행된 다음 수행된다. 따라서 M3에 대한 작업순서(Suf_i, I=N)는 두 가지 경우가 존재하며 다음과 같은 수식으로 표현할 수 있다.

$$Suf1 = t1 \rightarrow t4 \rightarrow t5 \rightarrow t6$$

$$Suf2 = t4 \rightarrow (t1 // t5) \rightarrow t6$$

각 작업순서(Suf)의 수행 시간(Tuf_i, I=N)은 다음과 같다.

$$Tuf1 = 13 + 20 + 7 + 6 = 46$$

$$Tuf2 = 20 + \max(13,7) + 6 = 39$$

2) M4에 대한 모델링 및 unfolding nets : M4은 그림 6과 같이 OP1과 OP2의 두 개의 작업에서 트랜지션 t3, t7 두 개의 공정에 이용된다. 그림 7은 M4에 대한 unfolding nets를 나타내고 있다.

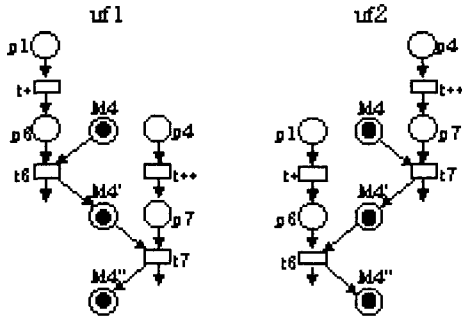


그림 6. M4 동작.
Fig. 6. Operation of M4.

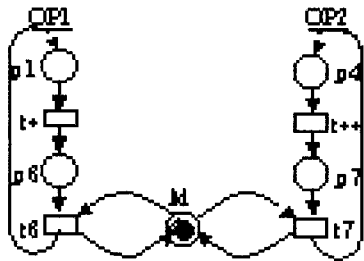


그림 7. M4에 대한 unfolding nets.
Fig. 7. Unfolding net of M4.

M4에 대한 작업순서는 그림 7과 같이 두 가지 경우가 존재하며 다음과 같은 수식으로 표현할 수 있다.

$$Suf1 = ((t^* \rightarrow t3) // t^{**}) \rightarrow t7$$

$$Suf2 = ((t^{**} \rightarrow t7) // t^*) \rightarrow t3$$

각각의 수행 시간은 다음과 같다.

$$Tuf1 = \max(23 + 11, 33) + 14 = 48$$

$$Tuf2 = \max(33 + 14, 23) + 11 = 58$$

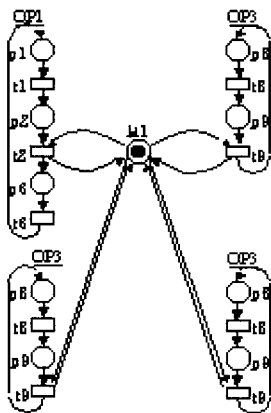


그림 8. M1 동작.
Fig. 8. Operation of M1.

3) M1에 대한 모델링 및 unfolding nets : M1은 그림 8과 같이 OP1 한 개의 작업과 OP3 세 개의 작업이 조합을 이루는 네 개의 작업으로 구성되어 있다. 그림 9은 M1에 대한 unfolding nets를 나타내고 있다.

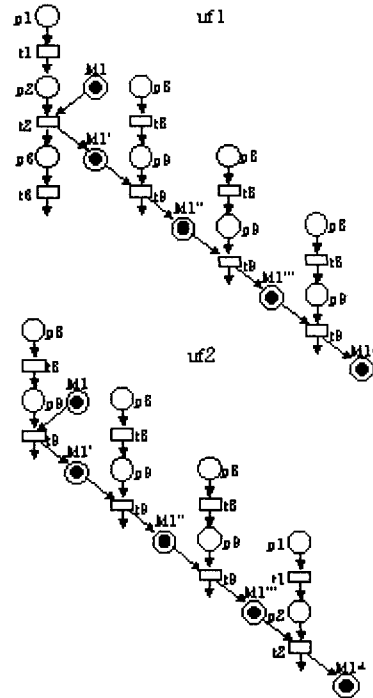


그림 9. M1에 대한 unfolding nets.
Fig. 9. Unfolding net of M1.

M1은 op1과 op3의 트랜지션 t2, t9에서 사용되며, 특히 op3은 op1이 한번 수행하는 동안 세 번 수행하게 된다. 따라서 M1에 대한 작업순서는 그림 9과 같이 두 가지 경우가 존재하며 다음과 같은 수식으로 표현할 수 있다.

$$Suf1 = ((((((t1 \rightarrow t2) // t8) \rightarrow t9) // t8) \rightarrow t9) // t8) \rightarrow t9$$

$$Suf2 = ((((((t8 \rightarrow t9) // t8) \rightarrow t9) // t8) \rightarrow t9) // t1) \rightarrow t2$$

각각의 수행 시간은 다음과 같다.

$$\begin{aligned} Tuf1 &= \max(\max(\max(23,4) + 7,4) + 7,4) + 7 \\ &= \max(\max(30,4) + 7,4) + 7 \\ &= \max(37,4) + 7 \\ &= 44 \end{aligned}$$

$$\begin{aligned} Tuf2 &= \max(\max(\max(11,4) + 7,4) + 7,13) + 10 \\ &= \max(\max(18,4) + 7,13) + 10 \\ &= \max(25,13) + 10 \\ &= 35 \end{aligned}$$

4) M2에 대한 모델링 및 unfolding nets : M1과 같은 방법을 적용하면 M2는 그림 10과 같이 OP2 한 개의 작업과 OP3 세 개의 작업이 조합을 이루는 네 개의 작업으로 구성되어 있다. 그림 10은 M2에 대한 unfolding nets를 나타내고 있다.

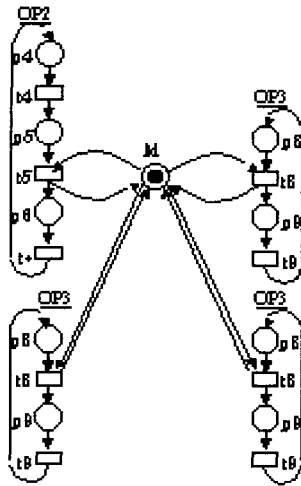


그림 10. M2 동작.
Fig. 10. Operation of M2.

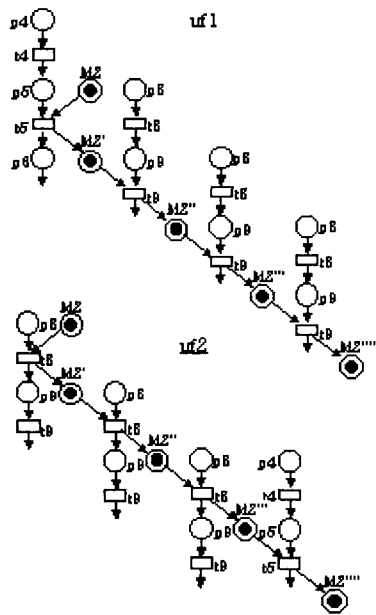


그림 11. M2에 대한 unfolding nets.
Fig. 11. Unfolding net of M2.

M2은 op2와 op3의 트랜지션 t5, t8에서 사용되며, 특히 M1과 같이 op3은 op2가 한번 수행하는 동안 세 번 수행하게 된다. M2에 대한 작업순서는 그림 11과 같으며 다음과 같은 수식으로 표현할 수 있다.

$$Suf1 = t4 \rightarrow t5 \rightarrow t8 \rightarrow t8 \rightarrow t8$$

$$Suf2 = ((t8 \rightarrow t8 \rightarrow t8) // t4) \rightarrow t5$$

각각의 수행 시간은 다음과 같다.

$$Tuf1 = 20 + 7 + 4 + 4 + 4 = 39$$

$$Tuf2 = \max((4 + 4 + 4), 20) + 7 \\ = \max(12, 20) + 7 = 27$$

3. 생산 비율을 직렬로 구성한 모델링

앞 절에서는 생산율을 달리하는 공정을 그 가중치만큼 병렬로 수행하는 방법을 생각하였는데, 이제는 이러한 작업형태를 직렬로 수행하는 경우의 모델을 생각할 수가 있으며 그 모델은 그림 12로 표현할 수 있다.

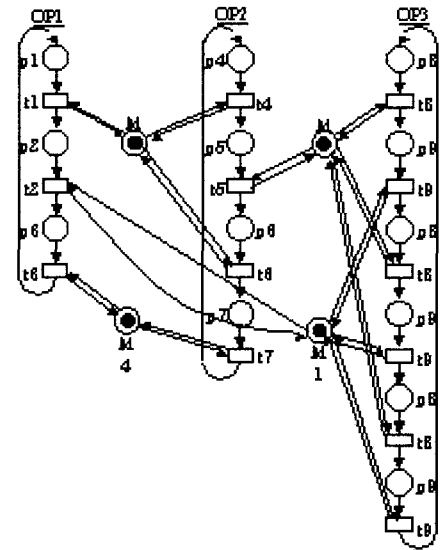


그림 12. 직렬로 표현한 모델.
Fig. 12. Serial model.

1) M2에 대한 모델링 및 unfolding nets : 직렬로 표현한 M2에 대한 동작은 그림 13과 같이 OP2 한 개의 작업과 OP3 세 개의 작업이 일렬로 구성된 두 개의 그룹으로 구성되어 있다. 그림 14은 M2에 대한 unfolding nets를 나타내고 있다.

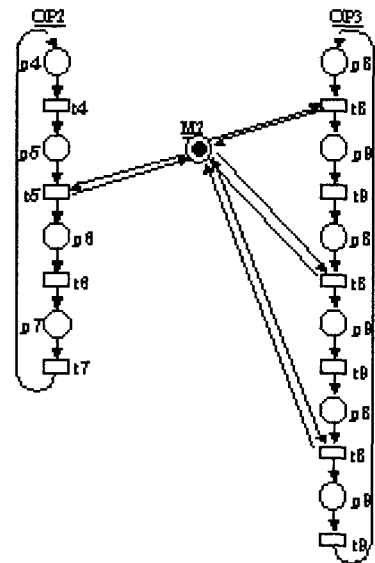


그림 13. M2 동작.
Fig. 13. Operation M2.

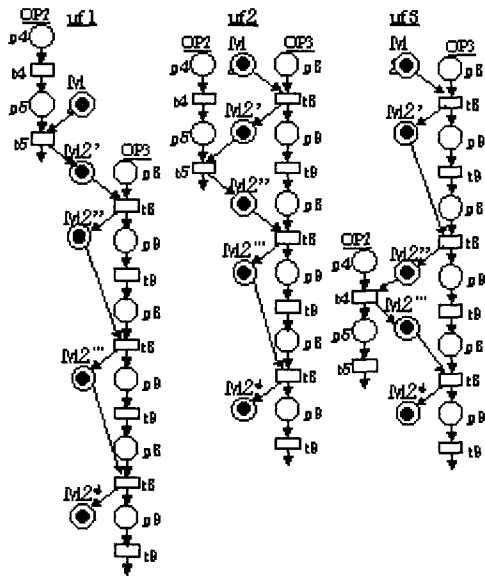


그림 14. M2에 대한 unfolding nets.
Fig. 14. Unfolding net of M2.

$$\begin{aligned} \text{Suf1} &= t4 \rightarrow t5 \rightarrow t8 \rightarrow t9 \rightarrow t8 \rightarrow t9 \rightarrow t8 \\ \text{Suf2} &= (t4//t8) \rightarrow (t5//t9) \rightarrow t8 \rightarrow t9 \rightarrow t8 \\ \text{Suf3} &= ((t4//(t8 \rightarrow t9 \rightarrow t8)) \rightarrow (t5//t9) \rightarrow t8 \rightarrow t9 \end{aligned}$$

각각의 수행 시간은 다음과 같다.

$$\begin{aligned} \text{Tuf1} &= 20 + 7 + 4 + 7 + 4 + 7 + 4 = 53 \\ \text{Tuf2} &= \max(20,4) + \max(7,7) + 4 + 7 + 4 = 42 \\ \text{Tuf3} &= \max(20,15) + \max(7,7) + 4 + 7 = 38 \end{aligned}$$

M2에 대한 모델링 및 unfolding nets

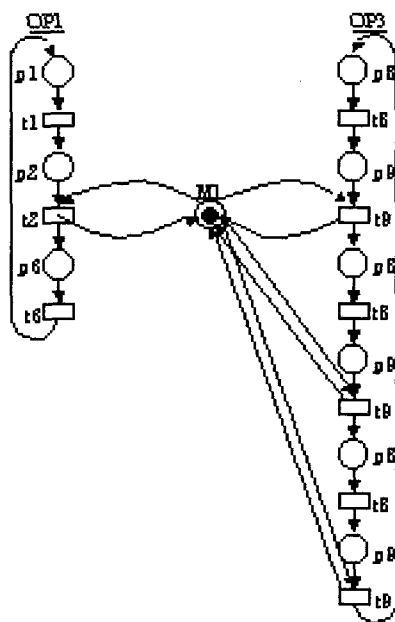


그림 15. M1 동작.
Fig. 15. Operation M1.

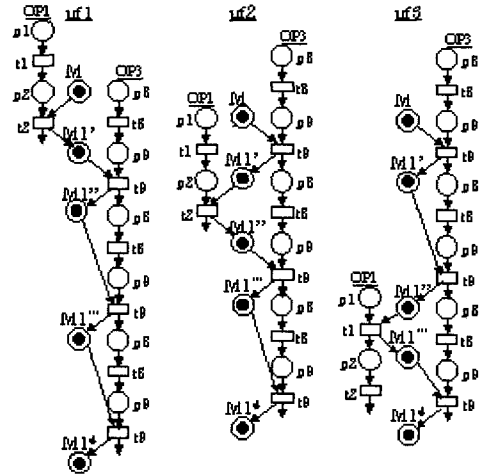


그림 16. M1에 대한 unfolding nets.
Fig. 16. Unfolding net M1.

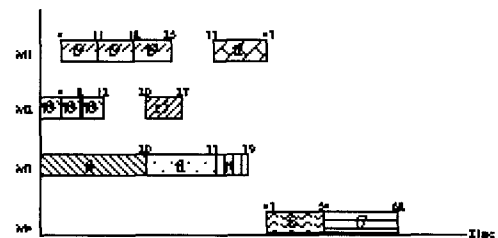
M1에 대한 동작과 unfolding은 그림 15 및 그림 16와 같으며, M1에 대한 작업순서에 대한 작업 시간은 다음과 같은 수식으로 표현할 수 있다.

$$\begin{aligned} \text{Suf1} &= ((t1 \rightarrow t2) // t8) \rightarrow t9 \rightarrow t8 \rightarrow t9 \rightarrow t8 \rightarrow t9 \\ \text{Suf2} &= (t1 // (t8 \rightarrow t9)) \rightarrow (t2 // t8) \rightarrow t9 \rightarrow t8 \rightarrow t9 \\ \text{Suf3} &= (t1 // (t8 \rightarrow t9 \rightarrow t8 \rightarrow t9)) \rightarrow (t2 // t8) \rightarrow t9 \end{aligned}$$

각각의 수행 시간은 다음과 같다.

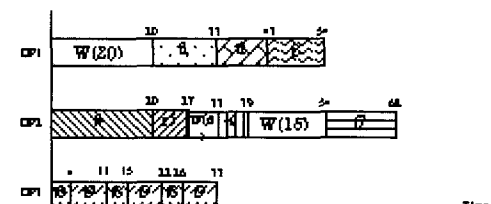
$$\begin{aligned} \text{Tuf1} &= \max(23,4) + 7 + 4 + 7 + 4 + 7 = 52 \\ \text{Tuf2} &= \max(13,11) + \max(10,7) + 7 + 4 + 7 = 41 \\ \text{Tuf3} &= \max(13,22) + \max(10,4) + 7 = 39 \end{aligned}$$

machine



(a) machine schedule

operation



(b) operation schedule

그림 17. 기기별 및 작업별 스케줄.
Fig. 17. Machine and operational schedule.

이상과 같이 그림 17의 machine operation을 생성하여 작업별 스케줄을 작성하면, 병렬로 구성하였을 경우 전체 작업 대기시간은 41이며, 직렬로 구성하면 58이 된다. 따라서, 병렬의 경우가 전체작업시간이 동일할 경우 대기시간에서 효율성이 높다고 할 수가 있다.

V. 결론 및 향후 연구과제

본 연구는 생산비율을 달리하는 WIP의 유연생산시스템 모델에 대해 분할형태의 패트리 넷 분석 방법을 이용하여 구조적 병행을 이루는 부분을 분해하고 unfolding을 이용하여 스케줄 분석에 관한 방법을 제시하였다. 특히 unfolding 기법을 이용하여 분석함으로써 시스템의 분석을 각 기기별로 구분하여 분석 할 수가 있었으며, 이를 종합화 하므로 한 시스템을 분석 할 수가 있었다. 이러한 개념은 패트리 넷을 여러 개의 서브네트로 구분하여 성질을 분석하고 이를 종합함으로써 전체적인 시스템의 성질을 분석 할 수 있다는 패트리 넷의 변형적 성질을 이용하여 유연생산 시스템에서의 스케줄링분석에 적용하여 보았다. unfolding개념은 주로 시스템의 성질 분석용으로 적용되어왔는데, 본 연구에서는 처음으로 세 가지의 작업과 네 개의 기기로 구성된 유연생산시스템에 적용하여 보았다. 분석 결과를 보면 생산비율이 다른 작업에서 작업처리를 생산 지용에 따라 병렬로 작업을 분산시켰을 경우가 직렬로 작업을 수행하는 경우보다 기기들의 대기시간을 보다 효율화 할 수가 있는 결론을 얻을 수가 있었다. 그러나, 본 연구에서는 각 과정에서 대기시간만을 주요 파라메타로 선정하였으며, 최적화에 관한 경우를 첨부하지 못한 한계점이 있어 다른 분석 기법과 그 효율성에 대한 비교 분석을 시행 할 수가 없었으며, unfolding을 이용하여 모델링하고 분석하는 기법을 제시하여 그 가능성을 보였다. 따라서, 다른 모델기법과의 비교분석을 위하여 최적화를 위한 파라메타의 추가 연구가 필요하다. 또한 보다 복잡한 시스템과 분산제어 시스템의 자원할당 문제에 대해서도 본 연구에서 시도한 개념을 적용하였을 때 발생할 수 있는 많은 unfolding nets를 자동으로 산출 할 수 있는 자동시스템의 개발 연구와 nets를 단순화 시키며 가장 효율적인 nets를 선택하는 변형 방법 및 알고리즘 개발 연구에 주력하고자 한다.

참고문헌

- [1] Berthomier B. and Diaz M., "Modelling and verification of time dependent systems using time petri nets," *IEEE Tr. Soft Eng.*, 17, pp. 259-273. 1991.
- [2] Holiday M. and Vernon M., "A generalized TPN model for performance analysis," *IEEE. Tran. On s/w, se-13*, pp. 1297-1310. 1987.
- [3] Zuberek W., "Schedules of flexible manufacturing cells and their timed colored petri netMod-

els," *In:Proceeding ICSYMC 1995*, pp. 2142-2147. 1995.

- [4] Hwang CH and Lee DI, "A concurrency characteristic in petri net unfolding," *In:Proceeding SMC 97*, pp. 4266-4273. 1997.
- [5] Lee DY and DiCesare F., "Petri net-based heuristic scheduling for flexible manufacturing," *In:Petri Nets in Flexible and Agile Automation*, ed. Zhou MC, Kluwer Aca. Pub., pp. 149-187. USA, 1995.
- [6] Carlier J. and Chretienne P., "Timed petri nets schedules," T R, Univ.P.et M., 1985.
- [7] Lee J., "Time composition problem in time petri nets", *In:Proceeding CIS' 97*, pp. 507-512. 1997.
- [8] T. Murate, "Petri nets: Properties, analysis an Applications," *Proceedings of the IEEE*, 77(4), pp. 541-580. April, 1989.
- [9] McMillan KL, "A thechnique of state space search based on unfolding," *In:Formal Methods in System Design 6(1)*, pp. 45-65. 1995.
- [10] McMillan. K, "A technique of state space search based on unfolding," *Formal Methods in System Design, 6(1)*, pp. 45-65. 1995.
- [11] Zuberek W. and Kubiah W., "Throughput analysis of manufacturing cells using timed petri nets," *In:Proceeding ICSYMC 1993*, pp. 1328-1333. 1993.
- [12] 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.
- [13] 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*, pp. 3081-3086. 1995.
- [14] 이우진, "Petri Net Slice를 이용한 Petri net 모델의 합성적 분석," 한국정보처리학회 추계 학술발표논문집 제6권 제2호.



이종근

1974 숭실대학교 전산과 졸업동 대학원 수료. 1977 고려대학교 경영대학원 경영학과 수료. 1990 Univ. de Montpellier II 전산과 박사 수료. 1983~현재 창원대학교 컴퓨터공학과 교수. 창원대학교 전산소장, 자연대부학장역임. 관심분야: 패트리 넷, 성능분석, 정보보호, 스케줄링연구.