

유연생산시스템의 순환 스케줄 분석 알고리즘의 비교 분석 연구

송유진^o, 이 중근
정보시스템연구실/ 컴퓨터공학과
창원대학교, 641-773 창원, 경남
(syj, jklee@sarim.changwon.ac.kr)

A Benchmarking Study of the Cyclic Scheduling Algorithm For Analyze in FMS

Yujin Song, Jongkun LEE
LIS/Dept. of Computer Engineering
Changwon National Univ.

요약

순환 스케줄분석은 효율성과 경제성을 모두 제고하여야 하는 문제로 분석알고리즘의 구성이 쉽지 않다. 따라서, 서로 다른 요소들을 활용한 알고리즘들과, 새로운 분석알고리즘의 효율성 비교분석이 어렵다. 본 연구에서는 예제 모델을 통한 비교 알고리즘의 처리과정과 결과를 복잡도, 최적도와 노력도를 기준으로 비교 분석하였다.

1. 서론

유연생산시스템에서의 주요한 과제중의 하나는 경제성을 추구하는 최적화의 작업 환경을 구축할 수 있는 순환 스케줄링을 구축하는 것이다. 이러한 연구를 위하여 많은 학자들이 패트리 넬을 이용하여 순환 스케줄링 알고리즘을 제시하였다[1-7,9-11]. 그러나, 순환 스케줄링 알고리즘의 경우 크게 세 가지의 알고리즘으로 구분이 가능하다. Hillion [3] 은 작업의 수행 가능성 비율을 기초로 한 발견적 방법 (heuristic algorithm) 모델을 제시하였고 Korbaa [5] 는 기존의 작업 일정을 다시 재그룹화 하여서 작업 환경 WIP (Work in Process) 을 고려한 순환 알고리즘을 제시하였다. Unfolding 분석 알고리즘[10,11]은 본 연구팀과 EC Lill의 연구팀이 공동으로 제시한 분석 알고리즘으로 패트리 넬의 모델을 자원 공유의 기기를 중심으로 여러 개의 서브 넬으로 (BUC: Basic unit Control) 분리하여서 각각의 서브 넬을 unfolding 기법으로 분석하여 최적의 스케줄링을 얻은 후 이를 종합화 하는 기법이다. 추이적 행렬식[8]에서 BUC를 구하였으며, 자원공유 분석을 위한 unfolding 알고리즘을 제시하였다.

순환 스케줄링 알고리즘의 경우, 자원 공유의 특성을 이용하여 분석이 필요하므로, 발견적 방법으로 제시되어진 알고리즘과 unfolding 과 같이 패트리 넬을 기초로 한 분석 알고리즘으로 나눈다면, 이러한 각각 알고리즘에 대한 비교 분석이 어려운 실정이다. 즉 unfolding 알고리즘의 경우, 넬을 기초로 한 결과로 모델을 토대로 복잡도 산정이 가능하지만, 앞의 두 알고리즘의 경우 결과와 시간 이외에는 비교 분석 요건이 충분하지 못하다고 하겠다. 따라서, 본 연구에서는 이러한 특수성의 알고리즘들을 최적도와 복잡도 그리고 노력도의 비교 적도를 사용하여 비교 분석하고자 한다. 이러한 비교 분석 결과를 통하여

더 효율적인 순환 스케줄링 알고리즘의 제시에 기초가 되고자 한다.

본 논문의 구성은 2 장에서는 분석을 위한 예제 모델을 설명하고, 3 장에서는 각 순환 알고리즘의 작업과정을 설명한다. 4 장에서는 예제 모델을 중심으로 각 알고리즘에서 얻은 결과들을 비교하여 각 알고리즘의 특성들을 정리한다. 마지막으로 결론과 앞으로의 연구 방향 등을 기술한다.

2. 분석을 위한 예제 모델

본 장에서는 분석을 위하여 일반적으로 많이 적용되어진 예제[14]를 살펴 본다(그림 1). 두 개의 생산품목 (P₁ 와 P₂) 은 3 개의 기기: U₁, M₁ 와 M₂ 을 통하여서 작업이 수행된다. P₁ 는 3 개의 작업이 수행되는데: u₁(2 t.u.: 기본시간), M₁(3 t.u.) 와 M₂(3 t.u.) P₂ 는 두 개의 작업 수행 : M₁(1 t.u.) 과 U₁(2 t.u.)로 구성되어진다. 생산 비율은 E={3P₁, 2P₂}이다. 이 의미는 5 개의 생산을 할 경우생산비율이 각각 3/5 과 2/5 의 비율로 두 개의 품목이 생산된다는 것을 의미한다. 여기서, 두 개의 품목을 위한 치차는 각각 독립적인 것으로 가정한다. 각 치차는 부품을 하나씩 만 운반이 가능하다고 가정한다. 각 작업의 작업 수행 순서는 OS1 과 OS2 로 표현한다.

예제의 사이클 타임 CT 는 CT = 11 이며, WIP 의 값

$$\text{은[5]: } WIP = \sum_{\text{pallets type } i} \left[\frac{\sum_{\text{OS to be carried by } i} \text{Operating time}}{CT} \right] \text{ 에서}$$
$$WIP = \left[\frac{7+7+7}{11} \right] + \left[\frac{3+3}{11} \right] = 3$$

즉 최적의 필요한 치차의 수는 3 이 되며, 이는 최적화의 기준이 된다.

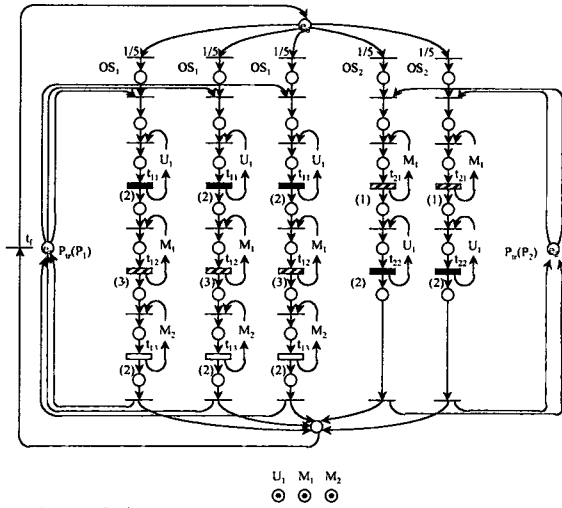


그림.1: 예제 모델

3. 순환 스케줄 분석 알고리즘

순환 알고리즘이란, 한정된 환경에서의 반복적 일정을 제시함으로써 생산의 최적화를 꾀하며 작업 시간의 단축을 꾀하는 작업 형태의 하나이다[11]. 이 장에서는 이러한 작업 형태의 최적화 스케줄링의 분석 알고리즘을 살펴 본다.

3.1 Hillion 알고리즘[3]

먼저, 각 작업별로 작업 편차도를 계산하여, 작업의 우선도를 통하여 최우선 작업과정을 구한다. 선별된 과정에서 기기의 자원공유문제를 비교하여 기기에서의 최적의 스케줄을 구한다. 다음의 공정을 선택하여 앞선 공정의 결과에 기인한 스케줄을 구하여 나간다.

3.2 Korbba 알고리즘[5]

먼저, 작업별 가능한 재그룹을 구성한다. 그리고 나서, 재그룹된 작업 공정을 통한 전체의 스케줄링 시간을 산정하여서 최적의 치차 수에 맞는 스케줄링을 찾는다. 만일 실패하면 재그룹의 내용을 변경하여 다음의 값을 찾는다.

3.3 Unfolding 알고리즘[10,11]

패트리 넬으로 구현되어진 모델에서 BUC라는 기기 중심의 자원공유 환경의 서브 넬으로 구분한다. 구분되어진 BUC의 자원공유도를 산출하여 가장 많은 공유도를 갖는 BUC를 중심으로 최적의 스케줄을 구하고, 계속해서 다음의 BUC를 통하여 스케줄을 구하여 나간다. 마지막으로 최적의 치차 수에 맞는 스케줄을 얻은 해법에서 찾아낸다.

4. 순환스케줄 알고리즘의 비교 분석

1) 최적도

예제 모델에 대한 처리 결과를 보면, 최적의 작업 운반차는 3대가 적합한데, Hillion의 경우에는 6대, Unfolding의 경우에는 4대 그리고 Korbba의 경우

에는 3대로 최적의 결과를 각각 얻었다. 따라서, 최적도의 경우 Korbba의 알고리즘이 가장 효율성이 높은 것으로 나타났으며 Hillion의 알고리즘이 가장 낮은 것으로 평가 되었다. 또한 unfolding의 경우 5개의 결과를 얻었는데, 그 중 2개가 서로 유사한 것으로 결과적으로 3개의 결과를 얻었다고 할 수가 있다(여기서는 1개의 결과만 표시한다).

2) 복잡도

unfolding 알고리즘의 경우 비교함수의 복잡도는 $n_1! * n_{i+1}!$ (여기서 n은 각 BUC의 자원공유의 수), 각 BUC에서 얻은 결과치가 동일할 경우 모든 계산되어진 결과를 다음 BUC에 적용하여야 하므로 그 복잡도는 기하급수적으로 증가하는 단점이 있다. 본 예제에서는 $240(5! : U1) + 120 * 4 + 4 * 6 = 744$ 의 비교수행량을 갖는다. Korbba의 알고리즘의 경우 재그룹 산출비교과정으로 9번, 재그룹 과정으로 30+4, 산출되어진 공정이 계산으로 52 따라서 95개의 산출식의 계산이 이루어진다. Hillion의 경우 비교공정이 5번에, 각 기기의 자원공유기여비율 3번, 각 공정에서의 기기 자원공유 수 만큼의 비교과정으로 120회, 2차에서는 120회 그리고 3차에서 6회를 통하여서 260회의 비교 공정이 나타난다. 각 알고리즘에서의 산출 수식이 다른데, Korbba의 알고리즘의 수식이 아주 복잡하며, Hillion의 경우 그리고 Unfolding의 경우 순으로 이어진다. 특히 Unfolding 알고리즘의 경우 모두 합산을 기초로 한 단순한 수식이어서 비교수행의 수는 많을지라도 산출 시간은 오히려 짧은 것을 알 수가 있었다.

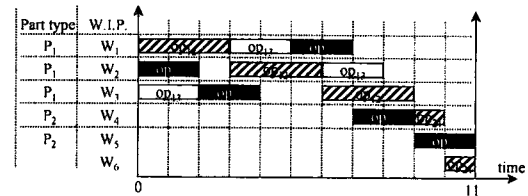


그림 2: Hillion의 알고리즘 결과

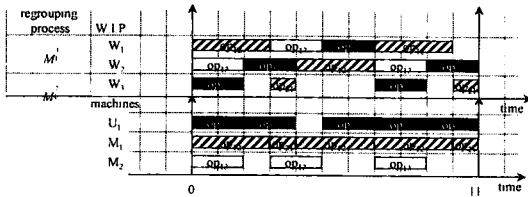


그림 3.: Korbba의 알고리즘 결과

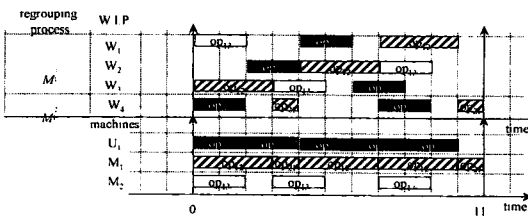


그림 4: Unfolding 알고리즘

3) 노력도

노력도의 경우 처리시간을 기준으로 분석하였는데, Hillion의 알고리즘이 빠르게 처리되어지는 결과를 얻었지만, 처음에 최적의 결과를 얻지 못하면 작업 흐름을 바꾸어서 새롭게 계산을 하여야 하는 관계로 많은 시간이 요구되어진다. Korbba의 알고리즘은 작업 처리과정을 재그룹화 하여야 하는 과정과 전체를 하나의 순환으로 연계하여서 분석하므로 Hillion의 처리보다 시간이 많이 걸린다, 특히 처음의 시도한 재그룹이 해당 얻기에 실패 하였을 경우, 다시 재그룹 하여서 계산하므로 상당한 시간이 요구된다. 예를 들어 9개의 기기와 7개의 작업과정에서 처리시간이 23시간이 소요되었다[5]. Unfolding의 경우, 기기의 자원공유 수에 기인하므로 작업의 수와는 무관하며, 9개의 기기와 7개의 작업과정에서는 3분의 처리 속도에 100개의 결과를[11] 보이므로 처리속도는 가장 효율적으로 나타났다.

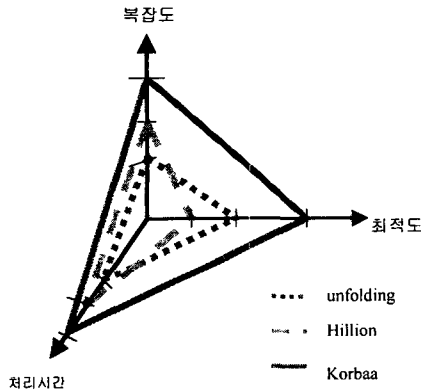


그림 5: 종합 비교도

Hillion은 작업의 수행 가능성 비율을 기초로 한 발견적 방법(heuristic algorithm) 모델을 제시하였고 그러나 이 모델들은 경제성을 추구하는데 최적의 작업 환경을 고려하지 않아 최적의 스케줄링이라고 정의할 수가 없다. Korbba는 기존의 작업 일정을 다시 재그룹화 하여서 작업 환경 WIP (Work in Process). 을 고려한 순환 알고리즘을 제시하였는데, 처음 그룹화 하여서 얻은 결과가 최적이 라면 가장 빠르게 최적의 값을 얻을 수가 있으나, 그렇지 못 할 경우, 그룹을 다시 구성하여서 처음부터 다시 계산하여야 하는 문제로 많은 시간이 소요 되어진다.

Unfolding 기법은 빠른 시간에 해답을 얻을 수가 있으며 다른 알고리즘이 한 개의 해답을 얻는 반면 여러 개의 해답을 얻을 수가 있는 장점이 있다. 그러나, Hillion 보다는 WIP에 가까운 작업 환경의 스케줄링을 얻지만, 항상 WIP의 결과를 만족하는 최적의 환경은 구하지 못하는 단점이 있다.

5. 결론 및 향후 연구

본 연구에서는 서로 다른 방법과 함수들을 사용한 순환 스케줄 분석 알고리즘을 비교 분석하였다. 비교 분석을 위하여서 처리과정에서의 노력도는 처리 시간분석으로, 최적도에서는 요구되어지는 치자의 수로, 복잡도에 대하여서는 요구되어지는 산술식과 이해 복잡도로 비교 분석하였다. 결론적으로 Korbba의 알고리즘이 최적도에서는 가장 앞서나, 노력도와 복잡도에서는 다른 알고리즘보다 효율성이 떨어지며, Hillion의 알고리즘의 경우, 노력도에서는 Unfolding보다 유리하나, 다른 면에서는 특히 최적도에서 취약성을 보였다, unfolding 알고리즘은 최적성에서는 Hillion보다 효율적이나, 항상 최적치를 만족하지 못하는 단점이 노출되었다, 또한 100개의 다른 해법을 얻었으나, 상당부분이 중복에 가까운 해법이 포함되어, 이러한 문제점들을 개선 보완 할 필요성이 있다. 향후 연구과제로는 이러한 알고리즘들의 비교를 수학적 표현들을 이용하여 정리 제시함으로 비교성능분석의 검증 기법의 제시가 필요하다.

References

[1]H.Camus,"Conduite de systèmes Flexibles de Production Manufacturière par composition de régimes permanents cycliques : modélisation et évaluation de performances à l'aide des Réseaux de Petri",Thèse de doctorat USTL 1,mars 1997
 [2]Zuberek W., Kubah W., "Throughput Analysis of Manufacturing Cells Using Timed Petri nets," In:Proceeding ICSYMC 1993, 1993, pp.1328-1333
 [3]H.Hillion,J-M Proth, X-L Xie,"A Heuristic Algorithm for Scheduling and Sequence Job-Shop problem",in proceeding 26th CDC 1987,pp.612-617
 [4]Korbba O,Camus H.,Gentina J-C,"FMS Cyclic Scheduling with Overlapping production cycles", in proceeding ICATPN'97,1997,pp.35-52
 [5] Korbba O,"Commande cyclique des systemes flexibles de production manufacturiere a l'aide des reseaux de Petri: de la planification a l'ordonnancement des regimes transtaires", Thèse de doctorat USTL 1 juillet 1998
 [6]Ofl 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
 [7] P. Richard," Scheduling timed marked graphs with resources : a serial method", accepted in proceeding INCOM'98, 1998
 [8] Jinghong LIU, Y.Itoh, I.Miyazawa, T.Seikiguchi,, "A Research on Petri nets Properties using Transitive matrix", in proceeding IEEE SMC99,1999,pp.888-893
 [9]Valentin C., "Modeling and Analysis methods for a class of Hybrid Dynamic Systems", in proceeding Symposium ADPM'94,1994,pp.221-226
 [10]Lee J.,Korbba O.,Gentina JC, "Modeling and analysis of Cycle scheduling using Petri nets unfolding", In: proceeding IEEE SMC'01, 2001,pp.2611-2616
 [11]Lee J., "Une méthode d'analyse d'ordonnancement des systems flexibles de production manufacturière utilisant le dépliage des réseaux de Petri", Thèse de doctorat EC Paris, mars 2002