

유한용량 자동생산 시스템의 효율적인 운용을 위한 시물레이션 기반 2단계 은행가 알고리즘(BA) 설계 및 성능분석

신희철¹ · 최진영^{1†}

Simulation-based Design and Performance Analysis of Two Phase Banker's Algorithm for Efficient Operation of Capacitated Automated Production Systems

Hee Chul Shin · Jin Young Choi

ABSTRACT

This paper suggests a two-phase look-ahead Banker's algorithm for efficient operation of capacitated automated production systems. The algorithm improves the ability of detecting safe states of the previous one by considering the possibility of processing each job to completion at once as well as partial movement between jobs. The improved performance of the proposed algorithm is testified by performing numerical experiment in terms of (i) detection rate of safe states and (ii) system throughput and verified by using paired t-test.

Key words : Capacitated Automated Production System, Banker's Algorithm, Safe State, Detection Rate, Throughput

요약

본 논문에서는 유한 용량 자동생산 시스템의 효율적인 운용을 위하여 기존에 제안된 은행가(BA) 알고리즘의 단점을 개선시킨 2단계 Look-ahead 은행가 알고리즘을 제안하였다. 제안된 알고리즘은 각 job 단위 별 전체 남은 공정의 처리 가능성뿐만 아니라 job들 간의 부분적인 이동 가능성을 동시에 고려하여 시스템 내부에 존재하는 job들의 완성 가능성을 결정함으로써 기존 알고리즘의 안전 상태 식별 능력을 향상시켰다. 수치 실험을 통해 제안된 알고리즘의 향상된 성능을 (i) 안전 상태 탐색율과 (ii) 시스템 Throughput을 기준으로 확인하였으며, paired t-test를 통해 검증하였다.

주요어 : 유한용량 자동생산 시스템, 은행가 알고리즘, 안전 상태, 탐색율, Throughput

1. 서론

최근 들어 사회 구조의 급속한 변화와 생활수준의 향상으로 소비자의 요구가 다양해진 현대 사회는 소비자가 주도하는 시장 체계로 변화되었다. 이러한 사회 환경의 변화는 산업 구조를 국내뿐만 아니라 전 세계적으로 무한 경쟁 체제의 경제 환경으로 전환시키는 계기가 되었다. 이로 인해 제조업체들은 글로벌 경쟁력을 갖추기 위해

은 적응력을 키워야 하며, 무엇보다도 생산성 향상과 원가 절감을 통한 이윤 창출을 위해 많은 노력을 기울여야 한다.

이를 위해 제조업체들은 자동화를 통한 생산 체제, 즉 자동생산시스템(Automated Production Systems)의 도입을 급격하게 추진하였다. 자동생산시스템이란 공업 제품의 설계나 생산에 있어서 컴퓨터에 의한 정보 처리를 도입하여 생산을 자동화하는 시스템으로 예전보다 생산성 및 품질 향상의 측면에서 많은 장점을 지니고 있다(Fanti 등, 2004). 그러나 이러한 자동생산시스템도 긴 고장 시간과 값비싼 자원에 대한 낮은 이용률 등으로 인해 시스템 성능이 크게 저하될 수 있다. 특히, 버퍼 용량이 한정된 유한 용량의 자동생산시스템은 시스템 내에 존재하는 job들을 처리하기 위한 시스템 자원의 할당이 제약됨으로 인

* 본 논문은 석사학위를 위한 연구로 수행되었음(Shin, 2012) 접수일(2012년 9월 24일), 심사일(1차 : 2012년 11월 13일), 게재 확정일(2012년 11월 18일)

¹⁾ 아주대학교 산업공학과

주 저자: 신희철

교신저자: 최진영

E-mail; choijy@ajou.ac.kr

하여 시스템이 영원히 멈추게 되는 교착(deadlock) 상태로 빠지게 될 수도 있는데, 이는 시스템 운용에 매우 치명적이다(Reveliotis 등, 2001).

예를 들면, Fig. 1은 반도체 공정에서 대표적으로 활용되는 유한용량의 클러스터 톨로서 chamber1과 chamber2가 각각 2개씩의 유한 버퍼를 가지고 있는 경우를 나타낸다(Choi, 2012). 각각의 웨이퍼는 chamber1→chamber2→chamber3의 작업 절차를 거쳐서 처리되고, 각 chamber에서 작업이 끝난 웨이퍼는 중앙에 있는 한 대의 로봇에 의해서 다음 작업 단계를 위한 chamber로 운송된다. 이때 만일 chamber1에 첫 번째 job stage의 job이 한 개 있고, chamber2에 두 개의 작업이 존재할 때 새로운 작업을 chamber1에 로딩하게 되면 클러스터 톨에 존재하는 모든 작업이 더 이상 진행될 수 없는 시스템 교착 상태에 빠지게 되며, 효율적인 시스템 운용을 위해서는 이러한 상태가 발생하지 않도록 제어하는 것이 필요하다. 일반적으로 교착이 발생하지 않는 시스템 상태를 안전 상태(safe state)라고 하며, 이를 위한 시스템 제어를 structural control 또는 logical control 이라고 한다(Reveliotis, 2005).

지금까지 이러한 문제를 해결하기 위한 많은 연구가 진행되어 오고 있다. 그러나 자동생산시스템의 운영이 복잡할 경우 모든 교착 상태를 제어할 수 있는 최적 교착 제어 정책을 찾는 것은 매우 어려운 문제로 알려져 있으며, 대부분의 경우 sub-optimal한 교착제어 알고리즘이 활용된다. 대표적인 polynomial-kernel sub-optimal 교착제어 알고리즘은 다음과 같다(Reveliotis, 2005).

- Resource Upstream Neighborhood (RUN): 각각의 워크스테이션에 대해 그 upstream neighborhood에 있는 job들의 합이 그 워크스테이션의 버퍼용량을 초과하지

않는 시스템 상태만 허용하는 방법

- Resource Ordering (RO): 모든 워크스테이션들의 순서를 임의로 정하고 순서쌍을 만든 후 그 사이에 있는 job들의 합이 해당 순서쌍에 포함된 워크스테이션 버퍼용량의 합보다 작게 되는 시스템 상태만 허용하는 방법
- Banker's Algorithm (BA): 시스템에서 처리 중인 모든 job들을 마지막 job stage까지 처리하기 위하여 필요한 모든 자원들이 가용한 시스템 상태만 허용하는 방법
- Banaszak-Krogh Deadlock Avoidance Algorithm (BK-DAA): 워크스테이션들을 production zone으로 정의하고 해당 zone에 있는 job의 개수가 zone의 버퍼 용량의 합을 초과하지 않는 시스템의 상태만 허용하는 방법

이 중에서 BA 알고리즘은 다음과 같이 시스템 안전 상태를 확인하여 운용 제어 활동(예를 들면, job을 다음 작업 단계로 운송)을 제어한다. 먼저 고려 대상인 운용 제어 활동에 대해 그 결과로서 만들어지는 다음 시스템 상태를 고려한다. 만일 다음 상태에서 시스템에 존재하는 모든 job들을 마지막 job stage까지 처리하기 위하여 필요한 모든 자원들이 시스템 내에 가용하다면 해당 운용 제어 활동을 허용한다(Lang, 1999; Reveliotis, 2005).

이를 위해 BA 알고리즘은 각각의 job들을 현재의 job stage에서 마지막 job stage까지 진행시킬 수 있는지 확인하는데, 이 때 고려되는 job들 간의 순서는 중요하지 않다. 만일 어떤 job이 마지막 job stage까지 처리될 수 없는 상태라면, 즉 중간에 어떤 job stage에서 필요한 워크스테이션이 이미 다른 job에 의해 점유되고 있다면 그 job을 더 이상 고려하지 않고 다른 job으로부터 완성 가능성을 확인한다. 따라서 끝까지 처리될 수 없다고 판단되었던 job이라 할지라도 다른 job들이 완성되어 시스템을 빠져 나간 후 released된 워크스테이션을 다시 이용할 수 있게 된다. BA 알고리즘은 이러한 경우에 해당하는 시스템 상태까지도 허용할 수 있기 때문에 다른 sub-optimal 교착 제어 알고리즘 보다 식별할 수 있는 안전 상태 범위가 커질 수 있게 되어 시스템의 운용 효율성을 높일 수 있다.

한편 식별하여 허용 가능한 시스템 안전 상태의 크기는 시스템 운용의 융통성(flexibility)에 제약을 주며 결과적으로 시스템 성능에 직접적인 영향을 줄 수 있다(Choi, 2012). 예를 들면, Fig. 1에 있는 클러스터 톨에 대해 한번에 하나의 job만 처리하도록 운용 제어 정책을 적용한다면 그 때 허용되는 모든 상태는 안전 상태이지만 시스템의 성능은 매우 낮아진다. BA 알고리즘의 경우에도 시스템의 크기가 커지고 처리되는 job type이나 job route

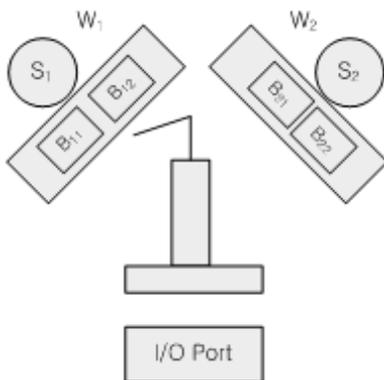


Fig. 1. An example of cluster tool(Choi, 2012)

등의 운용 조건이 복잡해질수록 안전 상태를 식별할 수 있는 능력이 떨어질 가능성이 많이 있다. 이는 BA 알고리즘의 안전 상태 식별 방법이 시스템 내에 있는 모든 job들의 완성 가능성을 각 job 단위 별 전체 남은 공정의 처리 가능성을 보고 판단하기 때문에 실제로는 안전한 상태가 안전하지 않은 상태(unsafe state)로 판정될 수도 있기 때문이다. 본 논문에서는 이러한 BA 알고리즘의 단점을 개선한 2단계 BA 알고리즘을 설계하고, 시뮬레이션을 통해 제안된 알고리즘의 우수성을 보였다. 시뮬레이션을 위한 시스템 운용 정책으로 일반적으로 많이 사용되는 dispatching rule들을 적용하였으며, 통계적 검증을 통해 고려된 dispatching rule들의 성능을 비교하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 유한용량 자동생산 시스템의 효율적인 운용을 위해 필요한 교착 제어와 성능 제어의 통합(Integration)에 관한 연구 문제를 정의한다. 3장에서는 이를 위해 본 논문에서 제안된 2단계 Look-ahead BA 알고리즘 설계 및 시스템 적용 방안을 설명한다. 4장에서는 수치 실험을 통해 제안된 알고리즘의 성능을 분석하며 마지막으로 5장에서 결론과 향후 연구 방향을 제시한다.

2. 연구 문제 정의

2.1 유한용량 자동생산 시스템 스케줄링 문제

본 논문에서 고려하고 있는 유한용량 자동생산 시스템은 제한된 버퍼 용량을 가진 m 개의 워크스테이션으로 구성되며, n 개의 job stage를 이용하여 제품을 생산한다. 이때 적어도 두 개의 job stage는 동일한 워크스테이션에서 처리가 되는 재진입 특성을 가지며, 이러한 특성은 $m < n$ 의 조건으로 표현될 수 있다.

이러한 유한용량 자동생산 시스템의 대표적인 예로서, Fig. 2는 인텔에서 제공하고 있는 Mini-fab 모델을 나타

낸다(Ramirez-Hernandez 등, 2009). 인텔 Mini-fab 모델은 모두 5개의 워크스테이션과 6개의 job stage로 구성되어 있다. 또한 이 모델은 다품종 생산, 배치 생산, 작업자의 작업, 트랜스포터, 기계 고장, 정기적인 유지 보수 등 일반적으로 시스템 운용 중에 발생할 수 있는 여러 가지 다양한 이벤트들을 모두 포함하고 있기 때문에 많은 연구에서 활용되고 있다. 특히 이 모델에서는 반도체 fab 라인의 주요 공정인 diffusion, ion implantation, lithography 공정을 포함하고 있으며, 각각의 작업은 start → diffusion → ion implantation → lithography → ion implantation → diffusion → lithography → exit의 단계를 거쳐서 완성되게 된다.

일반적으로 이러한 유한용량 재진입 자동생산 시스템의 효율적인 운용을 위해서는 각각의 워크스테이션에서 처리해야 하는 job stage가 두 개 이상일 때 어떤 버퍼에 있는 어떤 job을 먼저 처리해야 하는지를 결정하는 문제가 존재한다. 예를 들면 Fig. 2에서 워크스테이션 A/B는 버퍼 1과 버퍼 5에서 기다리고 있는 job들을 모두 처리해야 하는데, 시스템의 효율적인 운용을 위해서는 어떤 순서로 이 job들을 처리해야 하는가의 스케줄링 문제를 잘 해결해야 한다.

이 때, 각각의 워크스테이션은 제한된 용량의 버퍼를 가지고 있기 때문에 합리적인 스케줄링 정책이 적용되지 않을 경우 시스템 내의 모든 job들이 처리되지 못하고 정지되는 교착 상태에 머무르게 될 수도 있다. 또한 효율적인 스케줄링 정책은 교착 상태를 발생시키지 않는 동시에 시스템 전체의 성능을 최대화할 수 있어야 한다. 가장 간단한 교착 제어 정책으로 시스템 내부의 재공품 재고(Work In Process: WIP)이 언제나 1이 되게 하는 방법을 고려할 수 있지만, 이럴 경우 전체 시스템의 성능은 가장 낮게 되어 운용의 효율성이 매우 저하된다. 따라서 유한용량 재진입 자동생산 시스템의 효율적인 운용을 위해서는 교착

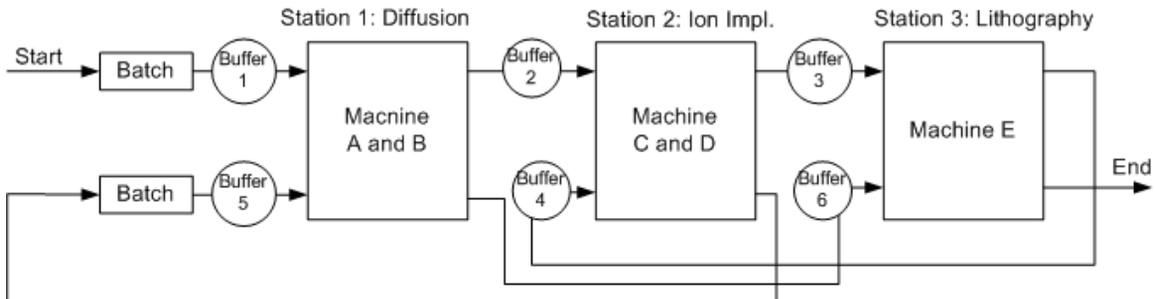


Fig. 2. Intel Mini-fab Model(Ramirez-Hernandez et al., 2009)

제어와 성능 제어를 동시에 고려하는 스케줄링 정책을 설계하는 것이 매우 중요하다.

2.2 교착제어 정책과 시스템 성능

일반적으로 인텔 Mini-fab과 같은 유한용량 자동생산 시스템의 교착제어와 성능제어를 동시에 고려하는 스케줄링 정책을 설계하기 위해서는 적용되는 스케줄링 정책에 따른 시스템 동작을 분석하는 것이 필요하며, 이를 위해서 유한 상태 기계(Finite state machine: FSM) 기법이 많이 활용된다. FSM이란 시스템 초기 유희상태에서 시작하여 모든 발생 가능한 시스템 상태를 표현한 방향성 있는 그래프로 정의된다(Cassandras 등, 1999).

그러나 일반적으로 시스템이 복잡할 경우 모든 시스템 상태를 표현하는 FSM의 크기가 기하급수적으로 증가하며, 결과적으로 모든 교착 상태를 제어할 수 있는 최적 교착 제어 정책을 찾는 것은 매우 어려운 문제로 알려져 있다. 따라서 1장에서 언급된 바와 같이 대부분의 경우 sub-optimal한 교착제어 알고리즘이 활용된다. 이 때 적용되는 교착 제어 알고리즘이 허용하는 안전 상태의 크기는 시스템 성능에 큰 영향을 미칠 수 있다.

예를 들어 Fig. 1에 있는 클러스터 틀에 대해 다음과 같은 교착 제어 정책을 적용한다고 가정하자.

- 교착제어정책 1 - 첫 번째와 두 번째 job stage에 있는 job의 수를 1이하로 유지
- 교착제어정책 2 - 첫 번째에서 세 번째 job stage에 있는 job의 수를 3이하로 유지하며, 두 번째 job stage에 job이 없으면 세 번째 job stage는 1이하로 유지

Fig. 3은 두 개의 서로 다른 교착 제어 정책을 적용할 경우 허용되는 안전 상태의 크기를 나타낸다. 작은 원으로 표시된 부분은 세 가지 job stage에 있는 job의 개수를 나타낸다. 예를 들어 (1 1 0)은 첫 번째와 두 번째 job stage에 한 개씩의 job이 있음을 의미한다. 이 때 적용된 교착제어 정책에 대해 허용되는 상태 범위 내에서의 최대 산출량(throughput)을 Discrete Time Markov Decision Process(DT-MDP)를 적용하여 계산할 수 있다(Puterman, 1994). 만일 모든 job stage에서의 처리 시간이 rate가 1인 지수 분포를 따른다고 가정할 경우 두 가지 교착 제어 정책을 적용한 최대 산출량은 각각 0.33과 0.48이다.

이러한 예를 통해서 알 수 있듯이 유한 용량 자동생산 시스템의 효율적 운용을 위해서는 안전 상태 식별 능력이 우수한 교착 제어 정책을 적용하는 것이 중요하며 이는

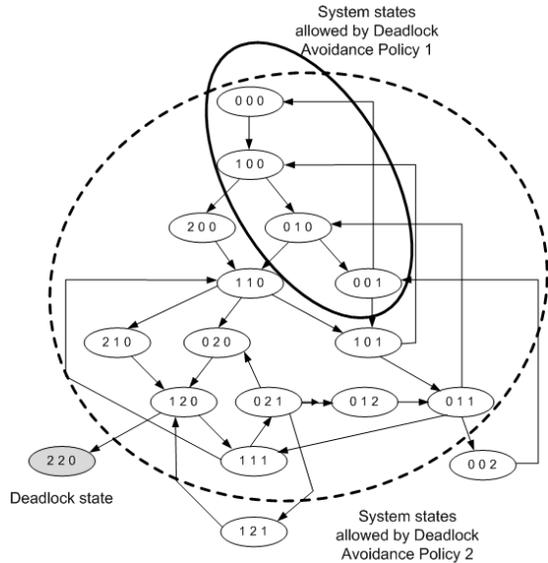


Fig. 3. Comparison of safe states for different deadlock avoidance policy(Choi, 2012)

시스템 성능에 큰 영향을 미칠 수 있다.

2.3 인텔 Mini-fab 시스템 예제

BA 알고리즘을 Fig. 2에 있는 인텔 Mini-fab 시스템에 적용할 경우 FSM은 매우 커지고 복잡해지며, BA 알고리즘의 안전 상태 식별 능력도 낮아지게 된다. 구체적인 예를 위해 Fig. 2의 인텔 Mini-fab 시스템의 구성 및 운용 세부 사항을 다음과 같이 가정한다.

- 각 Station에 있는 두 개의 버퍼가 물리적으로 동일
- Station 1의 버퍼용량이 2, Station 2의 버퍼용량이 3, Station 3의 버퍼용량이 3
- 각 Station의 Machine들은 이 버퍼들의 유희 공간을 공유

이러한 시스템 구성에 대해 BA 알고리즘을 적용하는 경우 시스템 상태 (2 3 1 0 0 0)은 다음과 같은 이유로 안전하지 않은 상태로 판정이 된다.

- 세 번째 job stage의 job을 네 번째 job stage로 이동하면 버퍼 용량 초과
- 두 번째 job stage의 job을 이동하는 경우 다섯 번째 job stage에서 버퍼 용량 초과
- 첫 번째 job stage의 job을 이동하는 경우 두 번째 job stage에서 버퍼 용량 초과

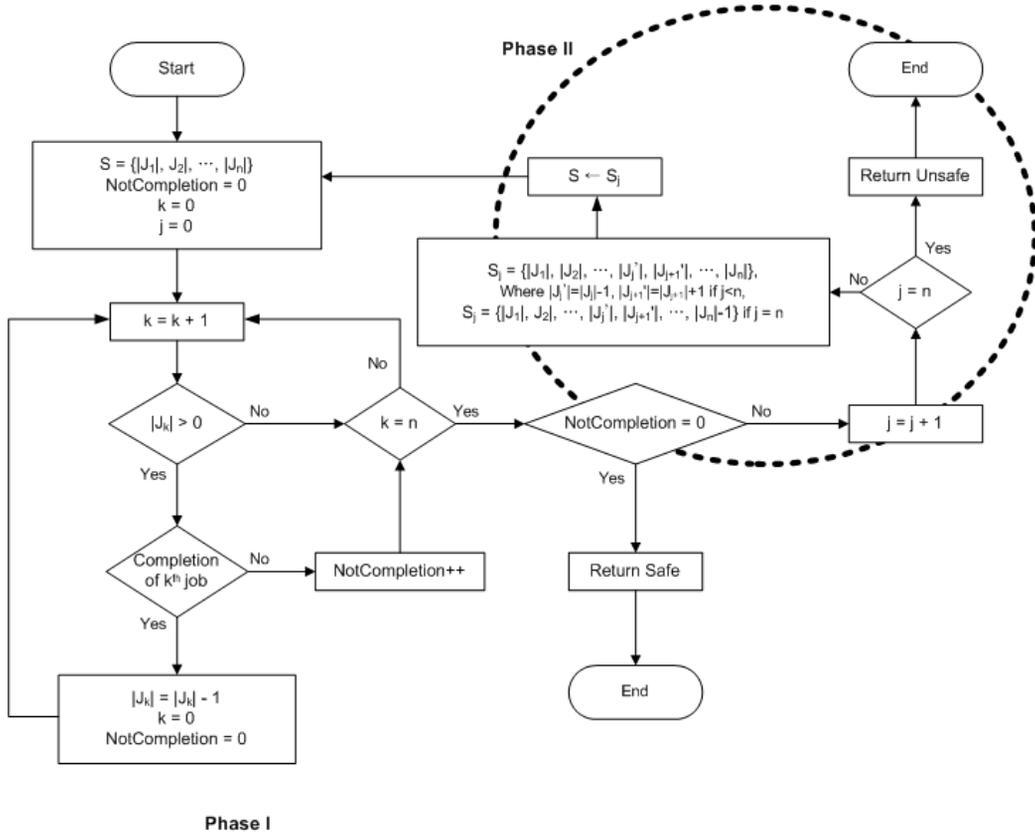


Fig. 4. Flow chart of 2-phase lookahead BA

그러나 이 시스템 상태에서 두 번째 job stage에 있는 job을 먼저 세 번째 job stage에 이동 시킨 후, 즉 (2 2 2 0 0 0)의 시스템 상태를 먼저 만든 후 BA 알고리즘을 적용하면 주어진 시스템 상태는 BA 알고리즘에 의해 안전 상태로 판정이 되며 실제로 이 시스템 상태는 안전하다. 따라서 BA 알고리즘에 의한 안전 상태 식별 방법의 성능은 이러한 방법에 의해 개선될 수 있다.

3. 2단계 Look-ahead BA 알고리즘 설계

3.1 알고리즘 설계

본 연구에서는 기존 BA 알고리즘의 안전 상태 식별 방법의 성능을 개선시킨 2단계 BA 알고리즘을 제안하였다. 제안된 알고리즘의 기본적인 아이디어는 시스템 내부에 존재하는 job들의 완성 가능성을 각 job 단위 별 전체 남은 공정의 처리 가능성을 보고 판단할 때 job들 간의 부분적인 이동 가능성을 동시에 고려하는 것이다. 이를 2단계

Look-ahead BA 알고리즘이라고 정의하며, 세부적인 절차는 Fig. 4와 같다. Fig. 4에서 사용된 기호의 의미는 다음과 같다.

- $|J_k|$ - job stage k ($k = 1, \dots, n$)에 있는 job의 수
- NotCompletion - 한 번에 끝까지 완성할 수 없는 job의 수
- $S_j - j$ ($j = 1, \dots, n$)번째 job stage를 변경시킨 시스템 상태

알고리즘은 크게 두 개의 단계(Phase)로 구성되는데 Fig. 4에서 실선으로 되어 있는 부분이 단계 1이고, 점선으로 표시된 부분이 단계 2를 나타낸다. 단계 1에서는 시스템 내부에 존재하는 job들의 완성 가능성을 각 job 단위 별 전체 남은 공정의 처리 가능성을 평가하며, 단계 2에서는 안전하지 않은 상태로 판정된 시스템 상태에 대해 job들 간의 부분적인 이동 가능성을 고려하여 안전성을 평가하게 된다.

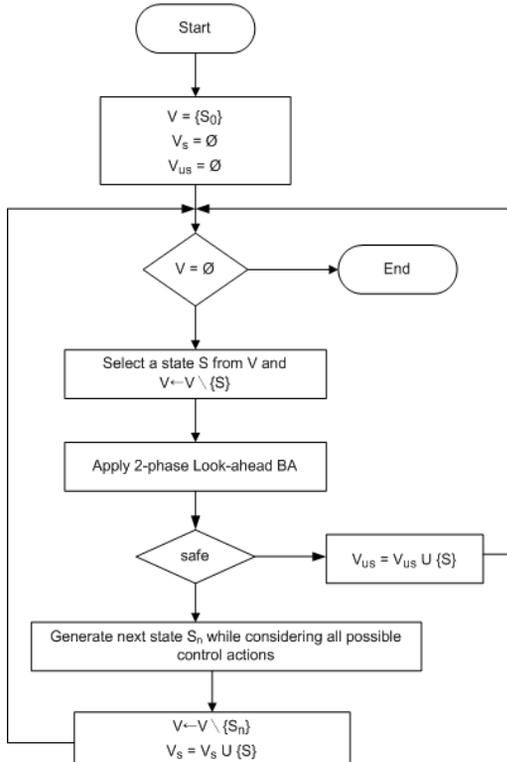


Fig. 5. Flow chart for FSM generation based on 2-phase lookahead BA

제안된 2단계 Look-ahead BA 알고리즘이 식별할 수 있는 대상 시스템의 전체 안전 상태를 나타내는 FSM을 생성하는 알고리즘은 Fig. 5와 같이 나타낼 수 있다. Fig. 5에서 V는 고려된 시스템 상태 집합, Vs는 알고리즘이 찾은 안전 상태 집합, Vus는 안전하지 않은 것으로 판단된 상태 집합을 나타낸다. 순서도의 중간에 있는 “2단계 Look-ahead BA 적용” 단계에서 Fig. 4에 있는 알고리즘을 적용하게 된다.

3.2 스케줄링 정책과의 통합 방안

교차제어 정책은 단순히 주어진 시스템 상태의 교차

여부를 판단할 수 있기 때문에 제안된 알고리즘을 유한 용량 자동 생산 시스템의 운용에 적용할 경우에는 실제 시스템의 운용을 위한 스케줄링 정책이 필요하다. 따라서 시스템의 운용 목적을 달성하기 위한 최적 스케줄링 정책을 비롯한 여러 가지 dispatching rule들이 고려될 수 있다.

그러나 일반적으로 유한 용량 자동 생산 시스템의 규모가 커지고 복잡해질수록 최적 스케줄링 정책을 찾는 것은 매우 어려운 문제가 된다는 것이 알려져 있으며(Reveliotis, 2005), 다양한 휴리스틱 정책이 활용되고 있다. 적용된 휴리스틱 정책은 주어진 시스템 상태에서 어느 한 가지 운용 제어 활동을 선택하게 되고, 적용된 교차 제어 정책은 그 결과로 발생하는 다음 시스템 상태에 대한 안전성을 확인하게 된다. 이 때 만일 다음 시스템 상태가 안전하지 않다면 해당 운용 제어 활동을 제외한 나머지 활동 중에서 하나를 적용된 휴리스틱 정책에 의하여 선택한다. 일반적으로 많이 쓰이는 휴리스틱 정책은 선입선출(First In First Out), 후입선출(Last In First Out), 선버퍼 선서비스(First Buffer First Serve), 후버퍼 선서비스(Last Buffer First Serve) 등의 dispatching rule들이 있다.

4. 수치 실험 설계 및 분석

4.1 수치 실험 설계

수치 실험을 통해 제안된 알고리즘의 성능을 평가하였다. 평가 대상 시스템은 인텔 Mini-fab 모델을 기반으로 워크스테이션의 수, 각 station의 버퍼 용량, 작업 순서, 각 job stage에서의 처리 시간을 다르게 하여 Table 1과 같이 4가지 시스템 구성으로 설계하였다(Shin, 2012). 구성 2와 구성 4는 구성 1과 구성 3에 대해 버퍼 용량과 작업 순서를 다르게 하여 다른 시스템 구성을 설계한 것으로서 시스템의 복잡성 증가에 따른 제안된 알고리즘의 행태를 분석하기 위한 것이다.

실험을 위해 제안된 알고리즘을 C언어로 구현하였으며, 2.33 GHz Core 2 Quad를 사용하여 실행시켰다. 제안된 알고리즘의 성능은 다음과 같이 두 가지 기준에 의

Table 1. Design of experiment for performance evaluation

| | # of workstations | Buffer capacity | # of job stages | Job sequences |
|---------|-------------------|-----------------|-----------------|---------------|
| Conf. 1 | 4 | (2, 1, 4, 2) | 6 | 1→2→3→4→2→3 |
| Conf. 2 | | (6, 5, 3, 4) | | 1→4→3→4→2→3 |
| Conf. 3 | 5 | (2, 3, 5, 1, 3) | 7 | 1→5→2→4→5→3→2 |
| Conf. 4 | | (3, 5, 4, 6, 3) | | 1→3→5→4→2→4→1 |

Table 2. Comparison of the # of safe states identified

| | # of total states | # of safe states | Current BA | Suggested BA | Rate of improvement |
|---------|-------------------|------------------|---|---|---------------------|
| | | | # of safe states identified / Identification rate | # of safe states identified / Identification rate | |
| Conf. 1 | 390 | 387 | 294 / 75.97% | 360 / 93.02% | 17.05% |
| Conf. 2 | 6,237 | 6,174 | 5,985 / 96.94% | 6,153 / 99.66% | 2.72% |
| Conf. 3 | 3,483 | 3,549 | 3,126 / 88.08% | 3,459 / 97.46% | 9.38% |
| Conf. 4 | 33,389 | 33,180 | 31,130 / 93.82% | 33,160 / 99.94% | 6.12% |

Table 3. Results of experiments for system throughput by considered deadlock avoidance policies

| | Current BA | | | Suggested BA | | |
|---------|------------|----------|----------|--------------|----------|----------|
| | FBFS | LBFS | FIFO | FBFS | LBFS | FIFO |
| Conf. 1 | 0.083612 | 0.083931 | 0.083757 | 0.090230 | 0.089736 | 0.089850 |
| Conf. 2 | 0.077570 | 0.077710 | 0.076740 | 0.080160 | 0.079120 | 0.081570 |
| Conf. 3 | 0.079777 | 0.079283 | 0.078437 | 0.082443 | 0.083461 | 0.081860 |
| Conf. 4 | 0.060080 | 0.059160 | 0.059440 | 0.063040 | 0.061400 | 0.060860 |

해 평가되었다.:

$$\text{기준1) 탐색율 (\%)} = \frac{\text{고려된 알고리즘이 찾은 안전 상태의 수}}{\text{전체 안전 상태의 수}} \times 100$$

$$\text{기준2) 시스템 Throughput} = \frac{\text{완성된 작업의 수}}{\text{실행시간}}$$

기준2)의 시스템 Throughput 평가를 위해서 FBFS, LBFS, FIFO 등의 dispatching rule을 적용하여 시뮬레이션을 실행하였다. 각각의 구성에 대해 총 30개의 다른 처리 시간을 갖는 시스템 운용 사례를 생성시켰으며, 주어진 작업에 대한 처리율은 0에서 1 사이의 소수 둘째 자리 수를 랜덤하게 발생시켰다. 또한 각각의 실행에 대해 실행 시간은 10,000으로 설정하였다. 구체적인 시뮬레이션 방법은 3.2절에서 설명된 내용을 기반으로 하였다.

4.2 실험 결과 및 분석

안전 상태 식별 능력에 대한 실험 결과는 Table 2와 같다. 각각의 구성에 대해 전체 시스템 상태 수, 전체 안전 상태 수, 기존 BA 알고리즘이 탐색 가능한 안전 상태 수, 제안된 알고리즘이 탐색 가능한 안전 상태 수를 평가하였다. 기준1)의 평가 항목에 대해 평균 8% 정도의 개선 효과를 확인할 수 있었다.

또한 전체적으로 실험 결과를 보면 기존 BA 알고리즘

Table 4. p-values of Paired t-test

| | FBFS | LBFS | FIFO |
|---------|----------|----------|----------|
| Conf. 1 | 0.000260 | 0.002023 | 0.000447 |
| Conf. 2 | 0.001902 | 0.000151 | 0.013747 |
| Conf. 3 | 0.000005 | 0.000992 | 0.000008 |
| Conf. 4 | 0.024283 | 0.017305 | 0.007369 |

을 적용했을 때의 안전 상태 탐색율은 75~97% 정도로 시스템의 구성에 따라 변동이 크게 분포되었다. 그러나 제안된 알고리즘은 93~99% 범위 내에서 비교적 일관적으로 높은 탐색율을 보이고 있다. 따라서 제안된 알고리즘이 일정하게 보다 높은 안전 상태 탐색율을 가지면서 손실되는 안전 상태를 감소시켜 시스템 성능 향상에 기여할 것으로 보인다.

두 번째 실험은 시스템 Throughput 향상에 대한 검증을 위해 실시되었으며 그 결과는 Table 3과 같다. 고려된 모든 시스템 구성과 dispatching rule들의 조합에 대해서 개선된 BA 알고리즘이 더 높은 Throughput을 보였으며 평균 7% 정도의 성능 개선을 확인할 수 있다. 이러한 실험 결과를 통계적으로 확인하기 위해서 다음과 같이 가설을 수립한 후 paired t-test를 실시하였다.

- $H_0: Th-Diff = 0$
- $H_1: Th-Diff > 0$

여기서 Th-Diff = 개선된 BA Throughput - 기존 BA Throughput을 나타내며, t 분포의 자유도는 29이다.

Table 4는 유의수준 $\alpha = 0.05$ 에 대해서 수행된 검정 결과를 나타낸다. 모든 경우에 대해 p-value가 유의수준에 비해 매우 작기 때문에 H_0 를 채택하지 않으며, 따라서 제안된 BA 알고리즘이 기존의 알고리즘에 비해서 높은 성능을 가짐을 알 수 있다.

5. 결 론

본 논문에서는 유한 용량 자동생산 시스템의 효율적인 운용을 위하여 기존에 제안된 BA 알고리즘의 단점을 개선시킨 2단계 Look-ahead BA 알고리즘을 제안하였다. 제안된 알고리즘은 각 job 단위 별 전체 남은 공정의 처리 가능성뿐만 아니라 job들 간의 부분적인 이동 가능성을 동시에 고려하여 시스템 내부에 존재하는 job들의 완성 가능성을 결정함으로써 기존 알고리즘의 안전 상태 식별 능력을 향상시켰다.

제안된 알고리즘의 성능 실험을 위하여 인텔 Mini-fab 시스템을 기반으로 워크스테이션 수, 버퍼 용량, 작업 경로 등을 다양화 한 4가지 종류의 시스템 구성을 설계하였다. 알고리즘의 성능은 (i) 안전 상태 탐색율과 (ii) 시스템 Throughput을 기준으로 평가하였다. 수치 실험을 통해 제안된 알고리즘이 기존 알고리즘 보다 향상된 성능을 가짐을 보였으며, paired t 검정을 통해 확인하였다.

이러한 결과는 기계의 버퍼가 많고 작업 경로가 매우 복잡하게 이루어지거나 규모가 큰 실제 산업 현장에서 시간적인 한계에 부딪혀 최적 교차 제어 정책을 찾기 어려운 경우에 기존 방법보다 효율적으로 최적에 가까운 안전 상태를 도출하는데 유용하게 활용될 수 있다. 따라서 시스템 운용의 융통성이 크게 확대되고, 이를 기반으로 시스템의 성능을 향상시켜 생산성 및 품질 향상 측면에서 많은 기여를 할 수 있을 것으로 기대된다.

참 고 문 헌

1. Cassandras, C. G. and Lafortune, S. (1999), Introduction to Discrete Event Systems, Boston, MA: Kluwer Academic Publishers.
2. Choi, J. Y. (2012), "Comparisons of PK-DAPs in the CRL Scheduling Problems: performance perspective", IJAMT (under review).
3. Fanti, M. P. and Chou, M. C. (2004), "Deadlock control methods in automated manufacturing systems", in IEEE Trans. on Systems, Man, and Cybernetics, vol. 34, no. 1, pp. 5-22.
4. Ramirez-Hernandez, J. A. and Fernandez, E. (2009), "A Simulation-Based Approximate Dynamic Programming Approach for the Control of the Intel-Mini-Fab Benchmark Model", In the Proceeding of the 2009 Winter Simulation Conference, pp. 1634-1645.
5. Lang, S. D. (1999), "An Extended Banker's Algorithm for Deadlock Avoidance", IEEE Trans. on Software Engineering, vol. 25, no. 3, pp. 428-432.
6. Puterman, M. L. (1994) Markov Decision Processes: Discrete Stochastic Dynamic Programming, New York: Wiley.
7. Reveliotis, S. A. (2005), Real-time Management of Resource Allocation Systems: A Discrete Event Systems Approach, Springer, New York.
8. Reveliotis, S. A., Lawley, M. A., and Ferreira, P. M. (2001), "Structural Control of Large-Scale Flexibly Automated Manufacturing Systems", in the Design of Manufacturing Systems, C. T. Leondes, Ed. Boca Raton, FL: CRC, pp. 4-1-4-34.
9. Shin, H. C. (2012), Design and Simulation-based Performance Analysis of two-phase Banker's Algorithm for Efficient Deadlock Control of Automated Production Systems with finite capacity, MS Thesis, Ajou University.



신 희 철 (hcshin@ajou.ac.kr)

2010 아주대학교 산업정보시스템공학부 학사
2012 아주대학교 산업공학과 석사 졸업

관심분야 : 스케줄링, 모델링&시뮬레이션



최 진 영 (choijy@ajou.ac.kr)

1991 한양대학교 산업공학과 학사
1993 KAIST 산업공학과 석사
2004 Georgia Tech 산업및시스템공학과 공학박사
2007~현재 아주대학교 부교수

관심분야 : 스케줄링(Multi-agent), 데이터 마이닝 최적화, Business Analytics