Journal of the Society of Korea Industrial and Systems Engineering
Vol. 34, No. 4, pp.189-196, December 2011.

시스템 특성함수 기반 평균보상 TD(λ) 학습을 통한 유한용량 Fab 스케줄링 근사화

최 진 영

아주대학교 산업정보시스템공학부

Capacitated Fab Scheduling Approximation using Average Reward TD(λ) Learning based on System Feature Functions

Jin Young Choi

Division of Industrial and Information Systems Engineering, Ajou University

In this paper, we propose a logical control-based actor-critic algorithm as an efficient approach for the approximation of the capacitated fab scheduling problem. We apply the average reward temporal-difference learning method for estimating the relative value functions of system states, while avoiding deadlock situation by Banker's algorithm. We consider the Intel mini-fab re-entrant line for the evaluation of the suggested algorithm and perform a numerical experiment by generating some sample system configurations randomly. We show that the suggested method has a prominent performance compared to other well-known heuristics.

Keywords : Fab Scheduling Problem, Actor-critic, Temporal-difference, Average Reward, Banker's Algorithm, Feature Functions

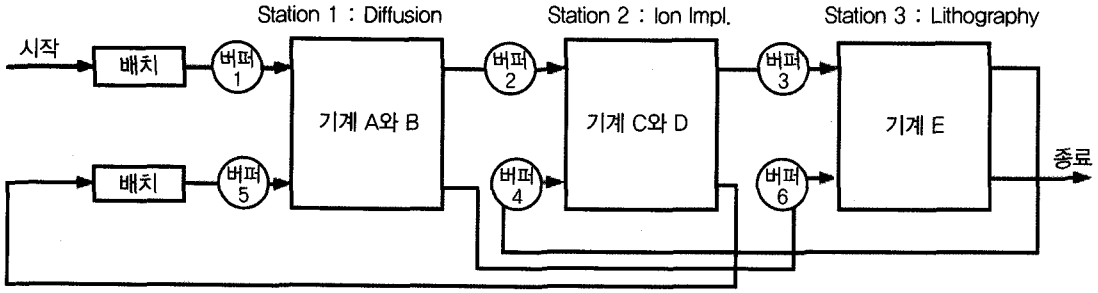
1. 서 론

최근 들어 반도체 및 LCD 산업을 비롯한 첨단 제조 시스템 분야에서는 모든 생산 공정이 점차로 완전 자동화 및 초소형화 되어 감에 따라 각 공정 단계에서 사용되고 있는 고가의 워크스테이션 등의 자원을 효율적으로 관리하고 스케쥴링하는 문제가 주요 이슈로 제기되고 있다. 그러나 생산 공정의 복잡도, 기술의 정밀도 및 버퍼와 서버 자원의 유한성 등으로 인해 제품의 생산성(throughput) 또는 사이클 타임 등이 고려된 전체 시스템의 성능을 최적화할 수 있는 효율적인 정책을 찾는 것은 매우 어려운 문제로 남아있다.

특히, 반도체 및 LCD Fab 공정에서는 하나의 완성된 웨이퍼를 생산하기 위해서 수십에서 수백 개로 구성된 공정단계가 필요하며 주요 생산 공정의 단계들이 여러 번 반복되어 수행되는 형태로 작업이 이루어진다. <그림 1>은 이러한 재진입 특성을 갖는 Intel mini-fab 모델을 나타낸다[14]. 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 있어 많은 연구에서 활용되고 있다. 특히 이 모델에서는 반도체 Fab 라인의 주요 공정인 diffusion, Ion implantation, lithography 공정을 포함하고 있으며 전체 작업은 start → diffusion → ion implantation → lithography → implantation → diffusion → lithography → exit의 단계를 거쳐서 완성되게 된다.

이러한 재진입 라인 시스템을 대상으로 한 효율적인 스케줄링 방법에 대한 연구는 지금까지 큐잉 네트워크, 추계적 모델링(Stochastic modeling), 시뮬레이션 등 다

논문접수일 : 2011년 10월 06일 논문수정일 : 2011년 12월 05일 게재확정일 : 2011년 12월 15일
† 교신저자 choijy@ajou.ac.kr



<그림 1> Intel Mini-fab 재진입 라인 모델

양한 연구 분야에서 많이 제안되어 왔다[5~10, 18]. 그러나 대부분의 연구에서는 워크스테이션의 버퍼 용량을 무한대로 가정하면서 시스템 성능에 대한 안정(stationary) 정책을 제시하고 있기 때문에 그 결과는 Fab 공정과 같은 유한용량 자동화 시스템에 그대로 적용하기에 적합하지 않은 제어 방법이 될 수 있다. 대표적인 예로 [12]에서는 버퍼 용량 제한이 없는 재진입 라인에서 지금까지 최적 생산량을 제공하는 스케줄링 방법으로 알려진 Last-buffer First-serve(LBFS) 정책이 유한용량 시스템에서는 더 이상 최적이 아님을 보였다.

이러한 결과는 버퍼 용량이 한정된 시스템에 대한 스케줄링 문제에서 시스템의 성능을 고려할 때 시스템 내부에서 움직이는 부품 이동의 논리적인 정확성(correctness)까지 보장해야 하기 때문에 발생한다. 이 때 부품 이동의 논리적인 정확성이란 버퍼 용량의 유한성으로 인해 시스템 내부의 모든 부품들이 움직이지 못하고 정체되는 현상이 발생하지 않도록 보장하는 것을 의미하며 이를 일반적으로 논리적 제어(logical control)이라고 한다[13]. 이러한 환경에서의 스케줄링 문제에 대한 복잡성(complexity)에 대해서는 아직까지 증명된 바가 없지만, 다계층(multi-class) 큐잉 문제와 논리적 제어 문제가 NP-complete이기 때문에 이 두 가지 문제를 동시에 고려한 유한 용량 Fab 스케줄링 문제는 더욱 어려운 문제임을 유추할 수 있다.

본 연구에서는 이러한 유한 용량 Fab 스케줄링 문제에 대한 효율적인 접근 방법으로 논리적 제어 기반 actor-critic 아키텍처 및 근사화 알고리즘을 제안하였다. 이 방법은 시스템 특성함수를 이용한 평균보상 temporal difference(TD)(λ) 학습 근사화 방법을 기반으로 한다. 이를 위해 먼저 [3]에서 제안한 시스템 특성함수를 이용하여 시스템 상태에 대한 상대가치함수(relative value function)[11] 예측에 필요한 일차 매개 함수를 구성하고, 이 함수의 계수를 찾기 위한 방법으로 [17]에서 제안한 평균보상 TD(λ) 학습 방법을 적용하였다. 이 과정에서 각 시스템 상태에 대해 예측된 상대가치함수를 이용하여 활동(action)을 선택할 때 그 결과로 정해지는 다음 상태의 논리적 정확성을 보장하기 위해 Banker's 알고리즘[13]을 이용하였다. 또한 <그림 1>의 Intel mini-fab 재진입 라인을 단순화하고 버퍼 용량을 유한개로 제한한 몇 가지 샘플 시스템을 설계하여 제안된 방법에 대한 성능 실험을 하였고, 기존에 많이 사용되는 다른 휴리스틱 방법과 비교함으로서 제안된 방법의 효율성을 보였다.

본 논문의 구성은 다음과 같다. 먼저 제 2장에서는 유한용량 Fab 스케줄링 문제에 대한 정의와 관련된 주요 연구 동향을 소개한다. 제 3장에서는 근사화 접근을 위한 NDP 모델링과 제안된 actor-critic 근사화 알고리즘을 설명한다. 제 4장에서는 제안된 알고리즘의 성능을 실험을 통해 분석하며 제 5장에서는 향후 연구 방향을 제시한다.

2. 문제 정의 및 관련 연구 동향

2.1 문제 정의 및 주요 이슈

본 연구에서 고려하고 있는 유한용량 재진입 라인이란 생산 시스템을 구성하고 있는 각각의 워크스테이션이 유한 용량의 버퍼와 유한개의 서버를 가지고 있을 때, 가공 대상인 각각의 job이 생산 시스템에 진입 후 모든 공정을 마칠 때까지 적어도 하나의 워크스테이션에서 두 번 이상 처리되는 재진입 특성 작업 공정을 갖는 것을 의미한다[2]. 이 때 각각의 워크스테이션에서는 처리해야 하는 job들에 대해서 가용한 버퍼와 서버를 어떻게 할당할 것인지를 결정하는 두 가지 의사결정 문제가 존재하며, 이는 전체 시스템의 성능에 직접적인 영향을 미치는 중요한 스케줄링 이슈이다. 특히 본 연구에서는 이러한 유한 용량 재진입 라인에서 생산성을 최대로 하기 위한 스케줄링 문제를 고려한다.

이러한 스케줄링 문제는 시스템의 규모가 커짐에 따라 최적 해를 찾기가 더욱 어려워지며, 이를 위한 근사화 방법이 절실히 요구된다. 지금까지 연구된 대표적인

근사화 방법은 시스템의 각 상태에 대한 가치함수를 예측하여 의사결정에 활용하는 방법인데, 이를 위해 look-up 테이블을 이용하는 방법과 매개변수 함수를 이용하는 방법이 있다[2, 16, 17]. 일반적으로는 시스템 규모가 커짐에 따라 동적인 모든 운용 조건을 표현하기에는 너무나도 많은 상태와 활동들이 존재하기 때문에 look-up 테이블을 이용하는 방법 보다는 매개변수 함수를 이용하는 방법이 더욱 효율적인 것으로 알려져 있다.

그러나, 매개변수 함수를 이용하는 방법에 대해서도 (i) 매개변수 함수를 만들 때 어떤 시스템 특성함수를 이용할 것인지와 (ii) 매개변수 함수의 계수를 어떻게 정할 것인지가 매우 어려운 문제로 알려져 있다[2]. 특히 시스템 성능에 영향을 미치는 중요한 시스템 특성함수의 선정 방법은 적용되는 응용 분야에 따라 다를 수 있으며, 선정된 특성함수 집합에 대한 적합성을 평가하는 방법도 중요한 이슈가 될 수 있다.

2.2 주요 관련 연구 동향

지금까지 매개변수 함수를 이용하여 시스템 상태에 대한 가치함수를 근사화하기 위한 방법으로 NDP를 적용하는 방법이 많이 연구되고 있다[1]. 이 방법은 curse of *dimensionality* 때문에 직접적인 적용이 어려웠던 동적계획법의 개념에 대해 시뮬레이션 기반의 최적화 알고리즘의 아이디어를 적용함으로서 모든 시스템 상태를 생성하지 않고서도 근사 최적 해를 찾을 수 있는 장점을 가지고 있다[14]. 일반적으로 매개변수 함수를 이용한 NDP 기반의 근사화 방법은 앞에서 설명된 두 가지 기본 단계로 구성되며, 다음과 같은 관련 연구가 수행되어지고 있다.

첫 번째는 시스템 특성함수를 찾는 단계인데 이것은 일종의 데이터 압축 과정이라고 할 수 있으며, 이를 통해 응용 분야에 특정한 도메인 관련 지식을 선택된 특성 함수로 표현하는 것이 매우 중요하다. 반도체 Fab 라인 스케줄링 문제에 대해 매개변수 함수 구성을 위한 특성함수 선택 방법에 대해서는 [2, 3, 14] 등에서 제안되었다. [2]에서는 시스템 특성 함수를 각 job 단계에서 기다리고 있는 job, 처리 중인 job, 완성된 job 등에 대한 정보를 나타내는 시스템 기본 상태 함수와 이들 간에 존재할 수 있는 상호작용을 나타내는 합성(composite) 함수로 나누어서 표현하였다. 또한 실험을 통해 제안된 특성함수를 이용한 NDP 근사화 방법의 잠재적 성능이 기존에 알려진 휴리스틱보다 좋음을 보임으로서 제안된 특성함수 집합의 우수성을 보였다.

그러나, 이 방법은 시스템 규모가 커짐에 따라 제안된 특성함수의 수가 매우 증가하는 단점을 가지고 있기 때문에 [3]에서는 [2]에서 제안된 시스템 특성함수를 대상으로 주성분 분석(principal component analysis)[4]에 기반한 thresholding 방법을 적용하여 핵심 특성함수를 도출하였다. 도출된 핵심 특성함수는 각 job 단계에서 기다리고 있는 job과 완성된 job의 개수, 각 워크스테이션의 버퍼 점유율, 가용 용량 등의 기본 상태 함수와 이들의 합성 함수로 구성되며, 평균적으로 [2]에서 제안된 특성함수보다 50% 정도 적은 수의 함수를 갖는다. [14]에서는 <그림 1>의 Intel *mini-fab*에 대해서 각 job 단계에서 존재하는 job의 개수를 특성함수로 활용하였으며, 배치 생산, 작업자의 작업, 트랜스포터, 기계고장, 정기적인 유지 보수 등 시스템 운용 중에 발생할 수 있는 여러 가지 다양한 이벤트들을 고려하였으나 기본적으로 버퍼 용량의 제약이 없다는 가정 하에 모든 분석이 수행되었다.

두 번째는 매개변수 함수의 계수를 찾는 단계로서 앞에서 선정한 특성함수를 매개변수 함수로 활용하여 근사화 아키텍처를 수립한 후 그 가중치 값을 구하게 된다. 지금까지 근사화 아키텍처의 매개변수 가중치 값을 찾기 위한 방법으로 동적계획법 또는 NDP, 시뮬레이션, Reinforcement Learning(RL) 방법이 많이 고려되었다[1, 14, 15, 17]. 동적 계획법 또는 NDP 기반의 방법에서는 대상 문제를 이산 시간 마코프 의사결정 프로세스(discrete-time markov decision process : DT-MDP)[11]로 모델링 한 후 각 상태에서 적합한 활동을 찾기 위해 value iteration이나 policy iteration 방법을 적용하여 각 상태에 대한 상대가치함수나 각 활동에 대한 보상(reward)를 반복적으로 개선시켜 나간다[1]. 이 때 상대가치함수란 주어진 상태로부터 시스템을 운영할 때 각 상태에 대한 보수 값을 기준으로 전체 이익을 계산한 것과 단위시간 당 평균 보수 값을 기준으로 전체 이익을 계산한 것의 상대적인 차이를 의미한다.

시뮬레이션을 이용한 방법에서는 가중치를 임의의 초기 값으로 정한 후 그에 따른 가치함수를 예측하고 policy를 결정한다. 이를 기반으로 시뮬레이션을 수행하여 시스템의 각 상태에 대한 가치함수를 개선하고 다시 이를 기반으로 가중치 값을 계산하여 가치함수를 점차적으로 개선시켜 나가는 방법이다[14, 15]. RL 방법에서는 대상 시스템을 모의로 운영하면서 현재 시점과 다음 시점 간 보상의 차이를 계산한 후 이를 각 상태에 대한 가치함수를 예측하는데 활용하는 방법으로서 TD 학습법이 대표적이다[15]. 그러나, 지금까지 매개변수 함수를 이용한 NDP 기반의 근사 최적해 방법들은 거의 모두 무한용량 시스템을 가정하였으며, 유한 용량 재진입 라인 스케줄링 문제에 대한 근사화 관련 연구는 거의 이루어지고 있지 않은 실정이다.

3. NDP 모델링 및 근사화 알고리즘

3.1 시스템 특성함수 기반 NDP 모델링

DT-MDP로 모델링 된 유한 용량 재진입 라인 스케줄링 문제에 대해 NDP 기반 스케줄링 근사화 방법을 적용하기 위해 본 연구에서는 [3]에서 제안한 시스템 특성함수를 이용하여 일차 매개변수 함수 근사화 아키텍처를 다음과 같이 수립하였다.

$$h^*(i) \cong \hat{h}(i,r) = \sum_{k=0}^{K} [\phi(k)](i) \cdot r(k)$$

여기서 $h^*(i)$와 $\hat{h}(i,r)$은 각각 상태 i에서의 최적 상대가치 함수와 그에 대한 예측 값을 나타내며, 예측 값은 상태 i에 대한 시스템 특성함수 $[\phi(k)](i)$와 가중치 $r(k)$의 곱의 합으로 표현되었다. 이 때, $\phi(k)$는 $\phi(k) = ([\phi(k)](0), \cdots, [\phi(k)](|S|-1))^T$로 정의되며 S는 전체 시스템 상태 집합을 나타낸다. 특히 $k=0$인 경우에 대해서는 모든 상태에 대해 $[\phi(0)](i)=1$로 정의하여 특성함수들이 서로 종속적인 관계가 되어 편향된 결과를 나타내는 경우를 줄이고자 하였다.

이렇게 정의된 상대가치함수 근사화 아키텍처에 대한 가중치를 튜닝(tuning)하여 가중치 벡터 r에 대해

$$h^*(i) = \hat{h}(i,r^*) = \sum_{k=0}^{K} [\phi(k)](i) \cdot r^*(k), \ \forall i$$

조건을 만족시키는 r^*를 찾는 것이 NDP 방법의 궁극적인 목적이며, 이러한 조건을 만족시킬 수 있는 r^*를 이용하여 다음과 같이 상태 i에 대한 최적 활동 $u^*(i)$를 구할 수 있게 된다.

$$u^*(i) = \arg \max_{u \in U(i)} \left[G(i,u) - \mu^* \bar{\tau}_i(u) + \sum_{j \in S} \widetilde{p_{ij}}(u) \hat{h}(j,r^*) \right]$$

여기서 $U(i)$는 상태 i에서 선택 가능한 활동의 집합, $G(i,u)$는 상태 i에서 활동 u를 선택함으로서 얻을 수 있는 단일 단계(single-stage) 기대 이익, μ^*는 안정 상태에서의 최적 평균 생산량, $\bar{\tau}_i(u)$는 상태 i에서 활동 u를 선택한 후 머무는 기대 시간, $\widetilde{p_{ij}}(u)$는 상태 i에서 활동 u를 선택하여 다음 상태가 j가 될 상태 천이 확률을 의미한다[2].

예를 들면 <그림 1>에서 버퍼 1과 버퍼 5에 대기 중인 job이 있을 때 이를 하나의 시스템 상태라고 하며, 이 때 기계 A와 B를 포함한 워크스테이션에서는 처리할 job의 순서를 결정하는 2개의 선택 가능한 활동이 있다. 만일 버퍼 1에 있는 job을 먼저 처리하는 활동을 선택했을 경우 시스템은 다른 상태로 천이하게 된다. 이 경우에는 동시에 처리되는 job이 한 개이기 때문에 결과 상태가 유일하게 결정되지만 선택된 활동이 2개 이상의 job을 처리하는 경우라면 2개 중에 먼저 완료되는 job에 의해 다음 상태가 확률적으로 결정된다. 이 때 만일 처리 결과로서 시스템의 출력(생산)이 발생하면 해당 활동을 선택함으로서 기대 이익이 발생하게 된다.

그러나, 일반적으로 생산성과 같은 평균 보상을 고려한 DT-MDP 문제에 대해 이러한 조건을 만족시키는 r^*를 찾는 것은 매우 어려운 것으로 알려져 있으며[1], 이에 대한 근사화 방법으로 앞에서 설명된 여러 가지 기법들이 활용될 수 있다.

3.2 평균 보상 TD(λ) 학습

시스템 특성함수 기반 NDP 모델에서 매개변수 함수의 계수를 튜닝하기 위한 방법으로 본 연구에서는 평균 보상 TD(λ) 학습을 적용하였다. TD학습은 1988년 Sutton에 의해서 제안된 방법[15]으로 DT-MDP로 모델링 된 문제에 대해서 각 상태에 대한 상대가치함수를 매개변수를 활용한 일차식으로 표현하여 근사화 할 수 있는 방법을 제공한다. 이 때, TD란 현재 시간 t에서의 상태에 대한 가치함수와 시간 $t+1$에서의 상태에 대한 가치함수의 임시적인 차이를 나타내며, 이 차이가 계속 개선되는 방향으로 파라메타 값을 검색해 나가는 것이 이 알고리즘의 기본 개념이다.

평균 보상을 고려한 DT-MDP 문제에 대한 이 알고리즘의 수렴성은 다음과 같은 두 가지 조건을 만족할 때 보장됨이 [17]에서 증명되었다.

(i) 시스템 특성함수는 상호 선형 독립임
(ii) 모든 $r \in R^K$에 대해서 $\Phi r \neq 1$(R^K는 K차 실수 공간을 의미함).

본 연구에서 적용된 시스템 특성함수는 각각의 특성함수 간에 존재하는 분산 및 상관관계를 분석하여 전체 데이터의 변동을 잘 표현할 수 있는 주요성분을 도출하고 여기에 기여도가 큰 특성함수들을 이동범위(moving range) 기반 thresholding 알고리즘을 적용하여 중요도에 따라 선택하게 하는 주성분 분석 방법에 의해 선정되었기 때문에 위의 두 가지 조건을 모두 만족시킬 수 있다[3].

적용된 평균 보상 TD(λ) 학습 방법의 구체적인 내용은 다음과 같다. 만일 대상 시스템이 확률 P에 의해 상태 천이가 이루어지고 시간 t에서 상태 i_t에 있을 때 가중치 벡터 r_t와 평균 생산성 μ^*에 대한 예측 값이 주

어졌다고 가정하면, 상태 i_t에서 상태 i_{t+1}로의 천이에 대한 temporal difference d_t는 다음과 같이 정의된다.

$$d_t = g(i_t) - \mu_t + \tilde{J}(i_{t+1}, r_t) - \tilde{J}(i_t, r_t)$$

여기서, $g(i_t)$는 상태 i_t에서의 단위 시간 당 보상(생산성)을 의미한다.

한편 TD 알고리즘은 r_t와 μ_t의 값을 다음과 같이 개선시킨다.

$$\mu_{t+1} = (1-\eta_t)\mu_t + \eta_t g(i_t)$$
$$r_{t+1} = r_t + \gamma_t d_t \sum_{k=0}^{t} \lambda^{t-k} \phi(i_k)$$

이 때 γ_t와 η_t는 알고리즘의 step size, λ는 $[0,1]$의 범위를 갖는 파라메타이며 알고리즘의 수렴성을 위해 다음과 같은 조건에 의해 선택된다.

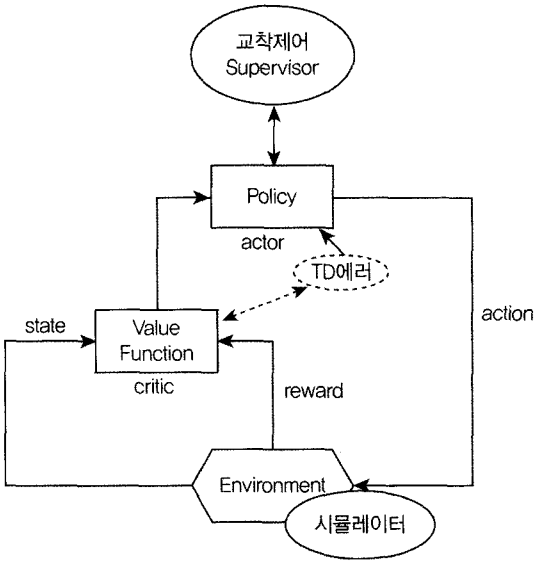
(i) γ_t는 $\sum_{t=0}^{\infty}\gamma_t = \infty$와 $\sum_{t=0}^{\infty}\gamma_t^2 < \infty$를 만족하는 deterministic한 양의 값을 가짐

(ii) η_t는 임의의 상수 c에 대해 $\eta_t = c\gamma_t$를 만족시킴

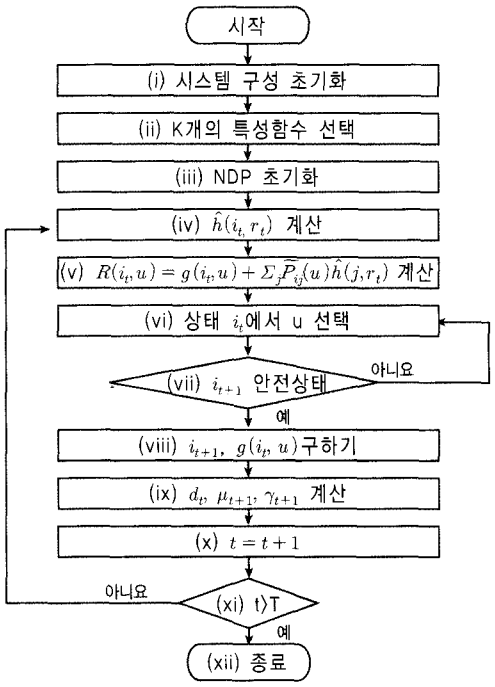
3.3 논리적 제어 기반 Actor-Critic 근사화 알고리즘 설계

유한 용량 재진입 라인의 스케줄링을 위해 평균 보상 TD(λ) 학습 알고리즘을 적용할 때 결과 상태에 대한 논리적인 정확성을 명확히 보장하는 것이 필요하며, 본 연구에서는 이를 위해 기존의 actor-critic 알고리즘에 추가적으로 교착 제어 기능을 연동하여 <그림 2>와 같은 논리적 제어 기반 actor-critic 아키텍처를 설계하였다.

제안된 아키텍처는 크게 actor, critic, 교착 제어 supervisor 및 시뮬레이터와 환경(Environment)로 구성된다. 이 때, actor는 현재 시간 t에서 학습된 가치함수를 이용하여 활동을 선택하는 역할을 하며 critic은 적용된 정책에 대해 다음 시간 $t+1$에서의 가치함수를 구하고 이에 기반한 temporal difference를 계산한다. 이 값은 앞에서 설명된 $\hat{h}(i,r)$ 값을 학습하는 데 이용된다. 교착 제어 supervisor는 actor에 의해 제안되는 정책을 적용한 결과 상태가 허용 가능한 상태인지를 확인하는 역할을 하며, 이를 위해 최적 교착 회피 방법, Banker's 알고리즘, Resource Upstream Neighborhood(RUN), Resource Ordering(RO) 등의 방법이 활용될 수 있다[13]. 본 연구에서는 Banker's 알고리즘을 적용하였는데 이 방법은 시스템에서 처리 중인 모든 job들을 마지막 job 단계까지 처리하기 위하여 필요한 모든 자원들이 가용한 시스템



<그림 2> 교착제어 기반 actor-critic 아키텍처



<그림 3> 논리적 제어 기반 근사화 알고리즘의 순서도

상태만을 안전(safe) 상태로 허용하는 방법이다. 이를 위해 각각의 job들을 현재의 job 단계에서 마지막 job 단계까지 진행시킬 수 있는 지 확인하는데, 이 때 고려되는 job들 간의 순서는 중요하지 않다. 선택된 활동에 대한 결과 상태 및 보상은 Environment에서 시뮬레이션을 통해서 정해진다.

<그림 3>은 이러한 내용을 바탕으로 한 논리적 제어 기반 actor-critic 근사화 알고리즘의 실행 순서도를 나타내며 각 단계에 대한 설명은 다음과 같다.

(i) 시스템 구성 초기화

먼저 고려되는 유한 용량 재진입 라인에 대한 기본적인 정보를 입력한다. 여기에는 시스템을 구성하는 워크스테이션의 수, 각 워크스테이션의 버퍼 용량, 완성된 제품을 만드는데 소요되는 job 단계의 수, 각 job 단계의 처리시간, 완성품을 만드는데 경유되는 워크스테이션의 순서 등이 포함된다.

(ii) K개의 특성함수 선택

[3]에서 제안된 K개의 효율적인 핵심 시스템 특성함수를 입력한다. 포함된 핵심 특성함수는 각 job 단계에 있는 job의 개수, 각 워크스테이션 버퍼 점유율, 각 워크스테이션 서버 가동 상태 등의 단순 특성함수와 이들의 상호작용을 나타내는 합성 특성함수로 구성된다.

(iii) NDP 초기화

NDP 기반 평균 보상 TD(λ) 학습을 적용하기 위해 t, r_t, γ_t, μ_t, i_t, $g(i_t)$, c, T 등 여러 가지 파라메타 값을 초기화 한다(T는 정해진 iteration의 반복수).

(iv) $\hat{h}(i_t, r_t)$ 계산

시간 t에서 시스템 상태가 i_t이고 가중치의 벡터 값이 r_t일 때 최적 상대가치 함수에 대한 예측치 $\hat{h}(i_t, r_t)$를 계산한다.

(v) $R(i_t, u) = g(i_t, u) + \sum_j \tilde{p}_{ij}(u)\hat{h}(j, r_t)$ 계산

(iv)에서 계산된 $\hat{h}(i_t, r_t)$ 값을 이용하여 시스템 상태 i_t에서 선택 가능한 모든 활동 u에 대해 보상 $R(i_t, u)$를 계산한다.

(vi) 상태 i_t에서 u 선택

(v)에서 계산된 $R(i_t, u)$를 이용하여 최대 $R(i_t, u)$ 값을 제공하는 활동 u를 선택한다.

(vii) i_{t+1} 안전(safe) 상태 확인

상태 i_t에서 선택된 활동 u를 적용하여 결정되는 다음 상태는 실행되고 있는 job 단계의 종류와 개수에 따라 결정된다. 본 알고리즘에서는 다음 결과 상태를 시뮬레이션에 의해 결정하며 이 때 i_{t+1}에 대한 논리적 정확성도 확인한다.

(viii) i_{t+1}, $g(i_t, u)$ 구하기

시뮬레이션으로 부터 결과 시스템 상태와 선택된 활동에 대한 보상을 계산한다.

(ix) d_t, μ_{t+1}, r_{t+1} 계산

Temporal difference d_t와 평균 보상 μ_{t+1}, 가중치 r_{t+1}을 계산한다.

(x) $t = t+1$

Iteration의 수를 하나 증가시킨다.

(xi) t < T

Iteration의 수가 미리 정해진 T보다 작은 경우에만 (iv)부터 (x)까지의 과정을 반복한다.

(xii) 종료

평균 보상 μ_t, 가중치 r_t 값을 반환하며 알고리즘을 종료한다.

제안된 알고리즘의 각 단계를 <그림 2> 아키텍처의 각 부분에 대응시키면 (i)부터 (iii)까지의 알고리즘 초기화 이후 critic 부분은 (iv)와 (v)에서 상대 가치함수를 계산하고, 이를 기반으로 actor와 교착제어 supervisor는 (vi)과 (vii)에서 안전상태를 보장하는 정책 u를 결정하게 된다. 이를 이용하여 environment와 simulator는 (viii)과 (ix)에서 시뮬레이션을 이용하여 시간 $t+1$에 대한 reward를 계산한다. 이 때 전반적인 알고리즘의 성능에 영향을 줄 수 있는 주요한 파라메타는 γ_t와 c값, 그리고 이들에 의해 결정되는 η_t값과 상대가치함수를 표현하기 위해 사용되는 시스템 핵심 특성함수이다.

4. 실험 설계 및 결과

4.1 실험 설계

제안된 논리적 제어 기반 actor-critic 근사화 알고리즘의 성능을 평가하기 위해 <그림 1>의 Intel mini-fab 모델을 기반으로 몇 가지 단순화된 재진입 라인을 설계하였다. 고려된 샘플 시스템에서는 <그림 1>의 작업 단계 중에서 주요 6가지 단계들이 3개의 워크스테이션에서 수행되는 것으로 가정하며, 이 때 세부적인 job의 처리 경로는 $W_1 \rightarrow W_2 \rightarrow W_3 \rightarrow W_2 \rightarrow W_1 \rightarrow W_3$($W_i$는 워크스테이션 i를 나타냄)으로 구성된다. 이러한 시스템에 대해 버퍼 용량의 크기를 몇 가지 경우로 나누어 <표 1>과 같이 7가지 종류의 실험모델 시스템을 구성하였으며, 이를 통해 3개의 워크스테이션이 가질 수 있는 상대적 버퍼 크기의 모든 경우가 고려되도록 하였다.

7가지 실험모델 시스템에 대한 실험은 각각의 실험모델에 대해서 6개의 job 단계 처리 비율(processing rate)

<표 1> 실험 대상 모델 구성

시스템 구성	버퍼 크기
실험모델 1	(2, 2, 2)
실험모델 2	(2, 3, 4)
실험모델 3	(2, 4, 3)
실험모델 4	(3, 2, 4)
실험모델 5	(3, 4, 2)
실험모델 6	(4, 2, 3)
실험모델 7	(4, 3, 2)

를 [1, 10]사이의 값으로 랜덤하게 바꾸면서 30개의 유한 용량 재진입 라인 스케줄링 문제를 생성하여 수행한 후 그 결과를 분석하였다. 이 때 각 단계에서의 처리 시간은 지수분포를 한다고 가정하였다. 또한 그 실험 결과를 비교·평가하기 위해 동일한 스케줄링 문제들을 기존에 많이 알려진 휴리스틱 방법인 First-in first-out, First-buffer first-serve, Last-buffer first-serve(이 방법은 많이 알려진 Shortest Remaining Processing Time 방법과 의미가 동일하다), Least-work next-queue 방법 등을 이용해 시뮬레이션 하여 생산성을 평가하였다. 그 이유는 스케줄링의 복잡성으로 인해 이러한 dispatching rules이 실제로 많이 적용되고 있으나 성능에 대한 한계점이 있으며 효율적인 스케줄링 알고리즘을 제안함으로서 이를 개선하고자 함에 있다. 시뮬레이션은 각각의 예제 문제에 대해 10,000시간을 1회로 100번의 반복 실험에 대한 평균 생산성을 계산하였으며, 이 때 전체 실험 모델에 대해 시스템 상태의 안전성을 보장하기 위한 교착 제어 알고리즘으로 Banker's 알고리즘을 동일하게 적용하였다.

일반적으로 제안된 알고리즘과 다른 휴리스틱과의 성능을 비교할 때 휴리스틱 방법의 성능을 최적해 성능과 비교하여 그 차이를 나타내는 % 오류를 다음과 같이 정의할 수 있다.

$$\%\text{오류} = \frac{TH_{optimal} - TH_{heuristic}}{TH_{optimal}} \times 100(\%)$$

그러나 이 방법은 각각의 실험모델에 대한 최적 생산성을 계산하기 위해서 Banker's 알고리즘에 의해 허용되는 모든 가능한 안전 상태 집합을 생성하고 이를 기반으로 한 DT-MDP 모델을 풀어야 하기 때문에 계산적으로 매우 비효율적이며 규모가 큰 시스템에 대해서는 최적의 정책을 찾는 것이 매우 어려운 문제(NP-complete)이다[13]. 따라서 본 연구에서는 %오류 대신 제안된 근사화 알고리즘과 휴리스틱의 성능 비교를 위해 평균 개선율을 다음과 같이 정의하고 이를 성능 평가의 지표로 활용하였다.

$$\text{개선율} = \frac{TH_{ac} - TH_{heuristic}}{TH_{heuristic}} \times 100(\%)$$

여기서 TH_{ac}와 $TH_{heuristic}$는 각각 제안된 actor-critic 알고리즘과 휴리스틱 정책에 의한 생산성을 나타낸다.

4.2 실험 결과 및 평가

먼저 수치 실험을 위해서는 제안된 논리적 제어 기반 actor-critic 알고리즘의 여러 가지 파라메터를 튜닝하는 과정이 필요하다. 본 연구에서는 각 실험모델에 대해서 알고리즘을 적용하여 실험한 결과 [17]에서 정의된 알고리즘 수렴을 위한 조건을 만족하면서 좋은 성능을 갖기 위한 파라메타 값으로 $c=0.5$, $\gamma_t = \frac{1}{t}$을 정하였다.

<표 2>는 7가지 실험 대상 모델에 대한 수치 실험 결과를 나타낸다. 각각의 실험 모델에 대해서 30개의 문제를 랜덤하게 생성하여 고려된 4가지 스케줄링 방법을 적용하여 생산성을 평가하고, 위에서 제안된 개선율의 평균값을 계산하였다. 표에서와 같이 제안된 알고리즘은 모든 실험모델에 대해 비교 대상인 모든 휴리스틱 방법보다 평균적으로 3%에서 15%까지의 개선된 성능 결과를 보였다. 이에 대한 유의성을 검증하기 위해 제안된 방법과 제일 좋은 성능을 보인 LBFS 성능의 차를 D로 정의하여 다음과 같이 paired t-test를 수행하였다.

$$H_0 : \mu_D = 0,$$
$$H_1 : \mu_D > 0.$$

통계량 t를 $t = \sqrt{n}\,\overline{D}/S_D$로 정의하고 계산한 결과 $t = \sqrt{209}(3.82)/(1.97) = 28.03 > t_{0.05,209} = 1.652$를 얻음으로서 제안된 방법에 의한 성능 개선 결과가 유의함을 통계적으로 확인할 수 있었다.

특징적인 것은 본 실험에서 고려한 <그림 1>의 Intel mini-fab 모델에 대해서는 LBFS 정책이 FBFS 정책보

<표 2> 수치 실험에 대한 개선율(%) 결과

	FBFS	LBFS	FIFO	LWNQ
실험모델 1	17.26	4.32	10.38	4.58
실험모델 2	15.39	3.27	9.73	3.97
실험모델 3	19.27	5.48	12.18	6.39
실험모델 4	13.33	2.73	8.74	3.18
실험모델 5	14.29	3.94	9.84	4.38
실험모델 6	12.27	2.37	7.49	2.58
실험모델 7	16.83	4.63	11.27	4.92
평균개선율	15.52	3.82	9.95	4.29

다 더 좋은 성능을 보였다. 그 이유로 본 실험에서 고려한 job의 처리 경로 구조가 경로 초입보다는 후미 부분에서 혼잡이 발생할 가능성이 많기 때문에 뒷부분의 job을 먼저 처리하는 것이 생산성에 더 유리한 정책이기 때문일 것으로 보인다.

5. 결 론

본 논문은 유한 용량 Fab 스케줄링 문제에 대한 효율적인 접근 방법으로 논리적 제어 기반 actor-critic 아키텍처 및 근사화 알고리즘을 제안하였다. 이를 위해 핵심 시스템 특성함수를 이용하여 시스템 상태에 대한 상대가치함수 예측을 위한 일차 매개 함수를 구성하고, 이 함수의 계수를 찾기 위한 방법으로 평균보상 TD(λ) 학습 방법을 적용하였다. 또한 Intel mini-fab 재진입 라인을 단순화하고 버퍼 용량을 유한개로 제한한 샘플 시스템을 설계하여 성능 실험을 하였고, 기존에 많이 사용되는 다른 휴리스틱 방법과 비교함으로서 제안된 방법의 효율성을 보였다. 본 연구 결과는 클러스터 툴 등의 재진입 특성을 갖는 반도체 생산 라인의 스케줄링 문제를 해결할 수 있는 기본 연구 결과로 활용될 수 있을 것으로 기대된다.

참고문헌

[1] Bertsekas, D. P. and Tsitsiklis, J. N.; Neuro-Dynamic Programming, Athena Scientific, 1996.

[2] Choi, J. Y. and Reveliotis, S. A.; "Relative value function approximation for the capacitated re-entrant line scheduling problem," *IEEE Trans. Autom. Science and Eng*, 2(3) : 285-299, 2005.

[3] Choi, J. Y. and Kim, S. B.; "Computationally efficient neuro-dynamic programming approximation method for the capacitated re-entrant line scheduling problem," *International Journal of Production Research* (accepted).

[4] Jolliffe, I. T.; Principal Component Analysis, Springer-Verlag, 2002.

[5] Kumar, P. R.; "Scheduling manufacturing systems of re-entrant lines," *in Stochastic Modeling and Analysis of Manufacturing Systems*, D. D. Yao, Ed. Berlin, Germanyy : Springer-Verlag, 325-360, 1994.

[6] Kumar, P. R.; "Scheduling semiconductor manufacturing plants," *IEEE Control Syst. Mag*, 14(6) : 33-40, 1994.

[7] Kumar, S. and Kumar, P. R.; "Fluctuation smoothing policies are stable for stochastic re-entrant lines," *Discrete-Event Dynam.* Syst., : Theory and Applicat., 6 : 361-370, 1996.

[8] Kumar, S. and Kumar, P. R.; "Queueing network models in the design and analysis of semiconductor wafer fabs," *IEEE Trans. Robot. Automat.*, 17(5) : 548-561, 2001.

[9] Lu, S. H. and Kumar, P. R.; "Distributed scheduling based on due dates and buffer priorities," *IEEE Trans. Autom. Control*, 36(12) : 1406-1416, 1991.

[10] Lu, S. H., Ramaswamy, D., and Kumar, P. R.; "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Trans. Semicond. Manuf*, 7(3) : 374-385, 1994.

[11] Puterman, M. L.; Markov Decision Processes : Discrete Stochastic Dynamic Programming, New York : Wiley, 1994.

[12] Reveliotis, S. A.; "The destabilizing effect of blocking due to finite buffering capacity in multi-class queueing networks," *IEEE Trans. Autom. Control*, 45(3) : 585-588, 2000.

[13] Reveliotis, S. A.; Real-time management of resource allocation systems, Springer, 2005.

[14] Rossetti, M. D., Hill, R. R., Johansson, B., Dunkin, A., and Ingalls, R. G.; "A simulation-based approximate dynamic programming approach for the control of the intel mini-fab benchmark model," In the proc. of the 2009 winter sim. conf., 2009.

[15] Sutton, R. S. and Barto, A. G.; Reinforcement Learning (An Introduction), MIT Press, 1999.

[16] Tsitsiklis, J. N. and Roy, B. V.; "Feature-based methods for large scale dynamic programming," *Machine Learning*, 22 : 59-94, 1996.

[17] Tsitsiklis, J. N. and Roy, B. V.; "Average cost temporal-difference learning," *Automatica*, 35 : 1799-1808, 1999.

[18] Wein, L. M.; "Scheduling semiconductor wafer fabrication," IEEE Trans. Semicond. Manufact., 1(3) : 115-130, 1988.