한국산학기술학회논문지
Vol. 11, No. 11 pp. 4174-4181, 2010

비안정적인 Rework 확률이 존재하는 제조공정을 위한 적응형 스케줄링 알고리즘

신현준[1*], 유재필[1]
[1]상명대학교 경영공학과

An Adaptive Scheduling Algorithm for Manufacturing Process with Non-stationary Rework Probabilities

Hyun Joon Shin[1*] and Jae Pil Ru[1]
[1]**Dept. of Management Engineering, Sangmyung University**

요 약 본 논문은 비안정적인 재작업 발생확률이 존재하는 제조공정을 위한 적응형 스케줄링 알고리즘을 제시한다. 본 논문에서 제안하는 하이브리드 Q-학습 알고리즘은 강화학습 기반의 Q-학습과 인공신경망을 결합한 알고리즘으로써 재작업확률이 불안정한 상황의 제조공정에 대해 학습을 통해 적응력을 가질 수 있도록 고안되었다. 제안 알고리즘은 평균지연시간을 척도로 그 성능을 평가하였고, 기존의 작업할당 알고리즘들과 다양한 실험 시나리오를 기반으로 비교함으로써 그 우수성을 보이도록 한다.

Abstract This paper presents an adaptive scheduling algorithm for manufacturing processes with non-stationary rework probabilities. The adaptive scheduling scheme named by hybrid Q-learning algorithm is proposed in this paper making use of the non-stationary rework probability and coupling with artificial neural networks. The proposed algorithm is measured by mean tardiness and the extensive computational results show that the presented algorithm gives very efficient schedules superior to the existing dispatching algorithms.

Key Words : Non-stationary Rework Probabilities; Dispatching rule; Q-Learning; Adaptive Scheduling; Artificial Neural Networks

1. 서론

재작업 (이하 rework)은 반도체나 박막트랜지스터 액정표시장치(TFT-LCD)와 같은 하이테크 산업뿐만 아니라 PCB 생산 등의 대다수 장치산업에서 제조공정의 효율성을 가늠하는 중요한 이슈가 되고 있다[2]. 제조업체가 경쟁력을 갖추기 위해 필요한 것은 고객의 요구사항을 만족시키는 것이고, 고객의 요구사항은 크게 두 부분으로 요약될 수 있다. 그 하나는 납기 내 적시 인도(delivery in time)이고 또 하나는 품질이다. 가공한 제품의 품질이 일정 기준에 미치지 못하게 된다면 이는 불량판정을 받아서 폐기되거나 rework을 통해 양품을 만들어야 하기 때문에 추가 발생하는 직접비도 무시할 수 없다. 따라서 납기준수 및 rework 최소화를 동시에 고려하는 효율적인 생산관리 기법의 개발이 필요하다.

작업의 투입순서를 결정할 때, 셋업시간과 납기 등의 주요 요인들과 함께 rework 발생 가능성을 효과적으로 우선순위에 반영할 수 있다면 위에서 언급한 목적 달성을 기대할 수 있을 것이다. 그러나 일반적으로 rework 발생 가능성은 시간의 흐름에 따라 예측 불가능하게 변화하는 비안정적인(non-stationary)형태를 갖고 있으며, 이것은 제품이 갖는 고유의 제조특성 치, 기계가 갖는 고유의 제조특성 치 그리고 이 둘의 조합으로 발생되는 제조특성 치들 간의 영향에 기인한다고 할 수 있다. 따라서 비안정적인 rework 발생 확률을 갖는 제조환경에서 효율적인 생산관리를 수행하기 위해서는, rework 발생 확률이

이 논문은 2008년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2008-331-D00694)
*교신저자 : 신현준 (hjshin@smu.ac.kr)
접수일 10년 09월 16일 수정일 10년 11월 03일 게재확정일 10년 11월 19일

미리 알려진 상수값이나 통계적인 분포를 갖는다는 가정이 비현실적이기 때문에, rework 발생 확률의 변화에 따라 작업의 투입정책을 수정해가는 스케줄링 알고리즘이 도입되어야 한다.

본 연구에서 다룰 문제 상황은 다음과 같다. Rework이 존재하는 워크센터(work center)에는 병렬기계의 형태로 구성된 흐름라인(flow line)이 있고 다양한 종류의 제품 타입을 갖는 작업들이 각 라인에서 가공된다. 만일 임의의 라인에서 가공을 마친 작업이 rework 판정을 받게 될 경우, 그 작업은 양품으로 통과될 때까지 재가공 절차를 거친다. 즉 rework이 발생하지 않을 때가지 작업이 기계군 중 하나에서 반복적으로 투입, 가공과정을 거치게 된다.

복잡한 제조공정에서 투입계획을 수립하는 방법은 아직 납기 또는 작업투입시점, 작업준비시간 등의 요인만을 고려하는 dispatching 규칙 및 단순한 휴리스틱에 의존하거나 rework 발생확률을 상수로 고정한다는 가정 하에 접근하는 것이 대부분이다. 하지만 본 연구의 대상이 되는 제조공정의 경우, 비안정적인 rework 발생 확률이 존재하므로 작업의 투입정책을 동적으로 변화시킬 수 있는 스케줄링 기법을 고안하는 것이 바람직하다. 따라서 본 연구에서는 강화학습(reinforcement learning) 기법을 이용하여 비안정적인 rework 발생 확률을 고려하는 효율적인 투입계획을 수립하는 적응형 스케줄링 알고리즘을 제안하고자 한다.

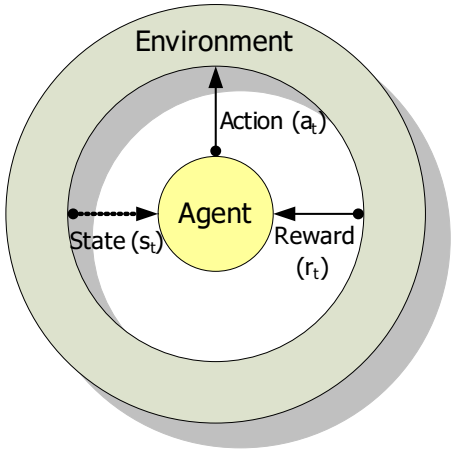
2. 선행 연구

Rework는 반도체 산업을 포함하여 많은 제조 산업에서 중요하게 고려되는 사항으로[1], 주로 비용의 최소화를 위해 최적의 생산배치크기나 재작업시기를 결정하는 문제 또는 최적의 rework 정책을 결정하는 것에 대한 연구가 진행되어 왔다. Liu[6]는 단일제품 생산 환경 하에서 단위시간당 평균이익을 최대화시키는 최적의 생산배치크기를 결정하는 방법을 제시하였고, Teunter & Flapper[8]은 제품을 가공하다가 일정 수량(N)의 rework 제품이 쌓이면 재가공작업을 시작하는 시스템에서 단위시간당 평균이익을 극대화시키는 N의 크기를 정하는 문제에 관하여 연구하였다. Kang et al.[2]는 rework 발생 확률을 상수로 고정한다는 가정 하에 rework을 고려한 dispatching 알고리즘을 제안하고 그 성능을 기존의 EDD(earliest due-dates), MS(minimum slack) 등의 dispatching 규칙들과 비교하였다.

Rework정책을 dispatching rule과 결합한 연구도 있다. Zargar[9]는 ‘mother lot'과 'child lot'의 개념을 도입하고 그들을 처리하는 방법에 따라 4가지 rework 정책을 제시하였다. 또한 실험을 통해 rework정책들이 cycle time에 미치는 영향에 관한 결과를 보여주었다. Kuhl & Laubisch[3]는 반도체 공정에서 dispatching rule과 rework 전략이 생산성에 가장 큰 영향을 주는 두 가지 요소라고 간주하고 5가지의 dispatching 규칙과 세 가지 rework의 전략 중 어떤 조합이 가장 높은 생산성을 얻을 수 있는지에 관하여 시뮬레이션을 통해 비교 분석하였다. Sha et al.[7]는 반도체 포토공정 (photolithography)의 dispatching을 통해 전체 공정이 운용된다고 보고 고객의 납기를 만족시키고 동시에 높은 제품품질을 얻기 위해서는 rework 전략과 dispatching 규칙을 적절하게 사용해야 한다고 말했다. 이 때 세 가지의 rework전략과 세 가지 dispatching 규칙의 조합으로 시뮬레이션을 수행하였다.

이상과 같이 기존 연구들을 고찰해본 결과 비안정적인 rework확률을 dispatching 알고리즘에 직접 반영하여 고안한 시도는 극히 드문 실정이며, 이는 일반 제조현장에서 rework과 관련한 데이터를 체계적으로 취합하여 작업 lot와 기계간의 세밀한 rework 발생 데이터를 정립하지 못하였거나 제조실행시스템 (manufacturing execution system)이 도입되었더라도 수율(yield) 데이터를 활용하기 위한 접근들이 미비했던 것으로 보인다.

본 논문은 다음과 같이 구성되어 있다. 3장에서는 본 논문의 주요 방법론인 강화학습의 기본 작동원리에 대해 설명하고 4장에서는 본 논문이 제안한 하이브리드 Q-학습 스케줄링 알고리즘에 대하여 기술하고 5장에서는 다양한 환경 하에서 실행된 실험 결과 및 분석을, 마지막으로 6장에서 결론 및 추후 연구 과제를 제시한다.



[그림 1] 강화학습에 의한 의사결정과정

3. 강화학습

3.1 강화학습 기법

본 연구에서는 비안정적으로 rework이 발생하는 환경하에서 최적의 적응형 스케줄링 알고리즘을 고안하기 위해서 강화학습 기법을 이용한다. 강화학습 기법의 전체적인 의사결정과정을 도식화하면 그림 1과 같다. 어떤 환경(environment)이 주어져 있고 이 환경을 이용하는 에이전트(agent)가 있다고 하면, 먼저 에이전트가 환경으로부터 어떤 상태(state = s_t)를 인식하게 된다. 이 때 에이전트는 특정 상태에 대한 여러 가지의 행동(action = a_t)을 취할 수가 있는데 그 행동 중 하나를 취하면, 그 행동에 대해 환경으로부터 보상(reward = s_t) 또는 벌점(penalty)을 받게 된다. 이러한 일련의 과정의 반복을 통해 에이전트는 각 상태에 대한 행동들 가운데 좋은 보상(또는 벌점)을 받는 쪽으로 점점 행동을 취할 가능성(확률)을 높여가게 된다. 이러한 방식으로 무한하게 반복을 하게 되면, 결국 하나의 상태에 대해 가장 좋은 행동 하나가 선택되어질 확률이 1.0에 가깝게 되어 상태와 행동이 일대일 대응관계로 되게 된다. 결론적으로 강화학습 기법이란 이러한 학습과정이 끝나면, 하나의 상태에 대해 취할 수 있는 여러 가지 행동 가운데 가장 성능이 좋았던 행동 하나만을 추출하여 최종적으로 하나의 상태에 대해 하나의 행동만을 취할 수 있도록 하는 기법이다[5].

3.2 Q-Learning

본 연구에서는 사용하는 기법은 강화학습 기법 중의 하나인 Q-학습이다. S를 environment 상의 상태들(states)의 집합, $A(s)$를 상태 $s(\in S)$하에서 가능한 행동들(actions)의 집합, γ를 지연된 보상(delayed reward)을 위한 할인율(discount rate)이라 할 때, Q-학습의 목표는 각 상태 s에 대해 기대되는 미래의 보상 값(expected future reward, 이하 Q-값)을 최대 또는 최소화하는 정책 $\pi : s \in S \rightarrow a \in A(s)$을 찾는 것이다. t시점에 s, a, π가 주어지면, Q-값을 산출하는 Q-함수는 다음과 같이 표현된다.

$$Q^{\pi}(s,a) = E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s, a_t = a, \pi\}$$

주어진 환경에는 다음을 만족하는 최적의 Q-함수($Q^*(s,a)$)와 정책($\pi^*(s)$)이 존재한다.

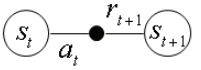
$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a),$$

$$\pi^*(s) = argmax_a Q^*(s,a)$$

이러한 최적의 Q-함수와 정책은 이론적으로 [그림 2]의 상태전이 과정 하에서 아래의 Q-값 갱신 과정을 무한히 반복함으로써 얻을 수 있다.

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma max_a(s_{t+1},a) - Q(s_t,a_t)]$$

여기서 $\alpha(0 < \alpha \leq 1)$는 학습율(learning rate)로써 확정적인 문제의 경우 α는 1의 값을, 확률적인 문제의 경우는 0과 1사이의 값을 갖는다. Q-학습은 초기의 Q-값으로부터 시작하여 '상태(state)파악→행동(action) 실행→보상(reward) 평가→Q-값 갱신'의 절차를 반복적으로 수행하게 된다. 이 때 학습과정에서 갱신되는 Q-값을 Q-테이블에 저장한다. 그러나 현실적인 크기의 문제의 경우, 상태변수가 증가함에 따라 보관해야 하는 Q-값의 양이 기하급수적으로 증가하여 Q-테이블 저장에 애로가 발생한다. 따라서 본 논문에서는 (s_t, a_t)와 해당 Q-값을 Q-테이블에 저장하는 대신 인공신경망(artificial neural network, 이하ANN)에 학습시켜 산출하는 하이브리드(hybrid) 알고리즘을 제안한다.



[그림 2] 1-step 상태전이

4. 제안 알고리즘

4.1 문제 정의

본 연구에서는 평균지연시간(mean tardiness; 이하 MT)을 최소화하기 위해 rework 가능성이 존재하는 상황에서 N개의 작업을 $M(m = 1,\cdots,M)$개의 병렬라인 형태의 기계에 스케줄링하는 알고리즘을 제시한다. N개의 작업은 같은 제품타입(product type)을 갖는 C개의 작업군으로 나뉘고, 이 때 작업군 J_c를 제품타입이 c인 작업들의 집합이라 한다. 작업군 J_c에 속한 작업들의 인덱스를 $j(j = 1,\cdots,N_c)$라고 했을 때, 이들 작업 j는 각각 납기(due date) d_j와 작업투입시점(release time) r_j를 갖는다. 각 작업 j의 가공시간(processing time) p_j는 작업들이 기계에서 처리되는 순서 및 가공 기계와는 독립적으로 주어지지만 작업준비시간은 작업들이 가공되는 순서와 속한 작업군에 따라 달라지는 순서의존적인 준비시간(sequence dependent setup times)을 갖는다. 여기서 순서의존적인 작업준비시간이란 작업군 J_c에 속한 작업 i의 가공을 마친 후 작업군 J_c 에 속한 작업 j를 준비하는 데 소요되는 시간을 s_{ic}라 할 때, $s_{ic} \neq s_{ci}$ 인 성질을 갖는

작업준비시간을 의미한다. 물론 작업 i와 j가 같은 제품군에 속한다면 $s_{cc'}=0$이다. 이 때 작업군 J_c에 속한 작업 j가 기계 m에서 가공을 한 후 불량 판정을 받아서 재가공을 할 확률은 P_{cm}이다.

의사결정 시점(epoch)인 t시점에 기계 m이 가공 중이던 작업 $i(i \in J_c)$의 가공을 완료한 후 유휴(idle)상태가 되어 투입작업으로 작업 $j(j \in J_c)$가 선택되었다고 하자. 현재 rework확률이 존재하기 때문에 작업 j는 기계 m에서 가공을 완료한 후 다시 가공을 해야만 하는 상황이 발생할 수도 있다. 만일 작업 j가 기계 m에서 가공을 완료한 후 rework판정을 받지 않을 경우에는 작업준비시간과 가공시간의 합인 $s_{cc'}+p_j$만큼의 시간이 소요되지만 rework 판정을 받을 경우에는 $s_{cc'}+p_j$뿐만 아니라 재가공에 필요한 시간이 더 소요된다. 따라서 의사결정 시점마다 납기지연 최소화와 rework 발생 가능성의 최소화를 동시에 고려하는 스케줄링이 필요하다.

본 논문은 이와 같이 정의되는 문제에 대한 솔루션으로 강화학습 알고리즘을 사용하고자 한다. 강화학습을 적용하기 위해서는 먼저 스케줄링 문제가 강화학습 문제 형태로 정의되어야 한다. 이 때 가장 기본이 되는 것은 상태(state)의 표현, 행동(actions) 방식, 그리고 보상(reward)함수의 정의이며 이어지는 절(4.2~4.4)에서 각 요소에 대해 자세히 설명하기로 한다.

4.2 상태(state) 표현

상태변수와 행동을 정의하는 것은 강화학습 기반의 스케줄링 알고리즘 개발에 있어서 가장 중요한 핵심요소이다. 상태변수는 작업과 기계정보를 포함한 시스템의 주요 특성을 설명함과 동시에 시스템의 상태 변화를 추적할 수 있어야 한다. 강화학습은 대상이 되는 시스템이 마코비안(Markovian) 또는 세미-마코비안(semi-Markovian) 속성을 갖는다고 가정한다. 따라서 의사결정 및 상태, 행동값은 오직 현재 상태에 따라 결정되며, 다음의(next) 상태와 보상은 주어진 현재 상태 및 선택된 행동에 의해서만 예측가능하다. 이러한 가정 하에서 에이전트와 환경(environment) 간의 상호작용이 바로 MDP (Markov decision process) 또는 SMDP(semi-MDP)가 되는 것이다. 결과적으로 본 논문의 스케줄링 문제를 강화학습 문제로 모형화하기 위해서는 4.1절에서 정의한 문제와 최대한 동일해지도록 상태변수를 도출하고 충실하게 표현하는 것이 필요하다.

본 연구에서는 의사결정 시점의 시스템 상태를 상태변수들의 벡터로 표현하고, 이 벡터는 총 $2N+3M+NM$개의 구성요소로 이루어진다. k번째 의사결정 시점(epoch)에서, s_k는 다음과 같이 정의된다.

$$s_k = [q_j (1 \le j \le N);\ e_j / max\{d_j\} (1 \le j \le N);\ T_m^0 / M (1 \le m \le M);\ T_m / M (1 \le m \le M);\ t_m / max\{s_{j_1 j_2} + p_{m j_2}\} (1 \le m \le M)(1 \le j_1, j_2 \le N);\ R_{jm} (1 \le j \le N)(1 \le m \le M)]^k$$

여기서,

q_j 작업 j가 queue에서 대기하고 있는 지 여부, 이진(binary) 변수로 다음과 같이 정의됨,

$$q_j = \begin{cases} 1 & \text{if } job\ j\ is\ waiting\ \text{for}\ processing \\ 0 & otherwise \end{cases}$$

e_j 작업 j의 납기까지의 여유도(slackness), $d_j - t_k$로 정의됨 (t_k는 k번째 의사결정 시점); 가공 완료되었을 경우 0의 값을 가짐,

T_m^0 기계 m에서 최근 완료된 작업,

T_m 기계 m에서 현재 작업 중인 작업 (단, 현재 기계가 유휴(idle)하다면 T_j=0),

t_m 기계 m에서 최종 셋업 후 경과된 시간,

R_{jm} 작업 j가 기계 m에서 작업 완료 후 rework판정 받은 회수이다.

예를 들어, 시스템 초기 상태 (즉 모든 작업이 queue에서 대기하고 있고, 모든 기계가 유휴한 상태) s_0는 다음과 같이 표현된다.

$$s_0 = [q_j = 1 (1 \le j \le N);\ e_j = d_j / max\{d_j\} (1 \le j \le N);\ T_m^0 / M = 0\ (1 \le m \le M);\ T_m / M = 0\ (1 \le m \le M);\ t_m / max\{s_{j_1 j_2} + p_{m j_2}\} = 0 (1 \le m \le M)(1 \le j_1, j_2 \le N);\ R_{jm} = 0 (1 \le j \le N)(1 \le m \le M)]^0$$

하나의 에피소드(episode)는 초기 상태로부터 종료 상태(모든 작업 가공완료)까지 스케줄을 생성해가는 과정을 뜻한다. 에피소드 시작단계의 시스템은 초기 상태 s_0이다. 기계들이 하나의 행동(action)을 취한다는 것은 가공할 작업을 선택하여 할당한다는 것과 같다. 그 후 임의의 기계가 가공중인 작업을 완료하면, 시스템은 새로운 상태 s_k로 변환된다. 에이전트가 의사결정 시점에 행동을 선택하면 시스템 상태는 과도기(interim) 상태 $\acute{s}$에 머무른다. 계속해서 어떤 기계든 가공중인 작업을 완료하면, 시스템은 다음 의사결정 상태인 s_{k+1}로 전이되고 에이전트는 r_{k+1}의 보상을 받는다. 하나의 에피소드는 이와 같은 절차를 종료상태 도달 시까지 반복한다.

4.3 행동(action)

본 연구에서는 에이전트의 학습능력을 높이기 위해,

특정 도메인(domain-specific) 지식을 사용하도록 한다. 스케줄링 시스템에 많이 사용되는 실시간 dispatching 규칙에는 EDD(earliest due-date), ATCS(apparent tardiness cost with setup), CR(critical ration) 등의 규칙이 있으나, 이러한 dispatching 규칙들은 rework 특성을 반영하지 않고 있어 그대로 사용하기에 부적합하다. 따라서 본 논문에서는 기존 CR 규칙에 rework 특성을 반영한 CRR(critical ratio with rework) 규칙을 고안하여 사용하도록 한다. CRR 규칙은 rework확률과 가공시간 및 셋업시간을 통해 얻어지는 가중치를 고려함으로써 대기행렬(queue)의 부하감소와 워크센터의 산출량(throughput) 증가를 통해 고객 납기만족을 목표로 한다.

Queue에서 대기하고 있는 모든 작업들을 대상으로 병렬라인 상의 임의의 기계(m)가 유휴상태(idle)가 되면 그 시점을 k번째 의사결정 시점(epoch) t_k라 한다. 이 때 CRR 규칙은 우선순위 함수($H_j(t_k,i)$)의 값을 구하고 가장 작은 값을 갖는 작업을 해당 기계(m)에 투입하도록 행동(action)을 결정한다.

$$H_j(t_k,i) = CRR_j(t_k,i) \times Weight_j(i)$$

여기서,

$$CRR_j(t_k,i) = \frac{d_j - t_k}{p_j + s_{c_i c_j}} \ ,$$

$$Weight_j(i) = \begin{cases} 1 & , if\ job\ j\ is\ new\ arrived\ one \\ \dfrac{p_j + s_{c_i c_j}}{\widehat{p_j} + \widehat{s_{c_i c_j}}} & , if\ job\ j\ is\ reentered\ for\ rework \end{cases}$$

i유휴상태가 된 시점(t_k) 바로 이전에 기계 m에서 완료된 작업의 인덱스,

$\acute{c}$ 작업 i가 속한 job class,

c 작업 j가 속한 job class,

$s_{\acute{c}c}$작업 i가 속한 job class($\acute{c}$) 가공완료 후, 작업 j가 속한 job class(c)를 기계 m에서 준비하는데 필요한 셋업시간(setup time),

$\overline{s_{\hat{c}c}}$작업 j가 속한 job class(c)가 기계 m에 투입되기 직전에 그 기계에서 가공을 완료한 임의의 job class($\hat{c}$)에 대한 평균작업 준비시간(average setup time),

$\widehat{s_{\acute{c}c}}$작업 j의 예상 셋업시간(expected setup time; EST) 즉, $(1-P_{cm})s_{\acute{c}c} + P_{cm}(s_{\acute{c}c} + \overline{s_{\hat{c}c}})$,

$\widehat{p_j}$작업 j의 예상 가공시간(expected processing time; EPT) 즉, $(1-P_{cm})p_j + P_{cm}(p_j + p_j)$ 이다.

요약하면 CRR 규칙은 워크센터 앞 대기행렬에 정상적으로 처음 진입한 작업보다 rework을 위해 재진입하는 작업에 보다 높은 가중치(weight)를 부여하게 된다. 이 때 해당 작업의 가공시간과 순서의존적인 셋업시간 그리고 납기까지의 상대 여유시간(slack)을 함께 고려해 준다.

4.4 보상(reward) 함수

보상함수는 현재 솔루션에 대해 행해진 행동의 영향을 값으로 산출하는 역할을 하며, 행동과 보상간의 관계를 나타낸다. 본 논문의 보상함수는 에이전트로 하여금 작은 값을 갖는 평균지연에 대해서는 큰 보상을 받도록 설계되었다. $\delta_j(t)$를 작업 j의 지연정보를 산출하는 함수라고 하면 다음과 같이 정의할 수 있다.

$$\delta_j(t) = \begin{cases} 0 & 0 \le t \le \min(c_j, d_j) \\ 1 & \min(c_j, d_j) \le t \le c_j \end{cases}$$

K를 전체 의사결정 시점의 수, $t_k (0 \le u \le K)$를 k번째 의사결정 시점, 그리고 r_k를 t_k시점에 에이전트로부터 받은 보상이라고 하면 r_k는 다음과 같이 표현된다.

$$r_k = \sum_{j \in SJ(t_{k-1})} \int_{k-1}^{k} \delta_j(\tau) d\tau + rw_k \ ,$$

$$rw_k = \begin{cases} \bar{p} & if\ \ rework \\ 0 & otherwise \end{cases}$$

여기서, $SJ_{t_{k-1}}$는 t_{k-1}시점에 대기하고 있거나 가공되고 있는 작업들의 집합을 뜻한다. rw_k는 t_k시점에 유휴한 기계에 할당된 작업이 rework판정을 받게 되면 평균가공시간($\bar{p}$)을 갖고, 그렇지 않을 경우 0의 값을 갖는다. 일반적인 강화학습에서는 보상값이 큰 값을 갖도록 학습이 이루어지지만, 본 연구의 대상문제는 납기지연평균의 최소화를 목적으로 하고 있기 때문에 위 r_k는 작을수록 좋은 페널티(penalty)의 의미로 인식하여야 한다.

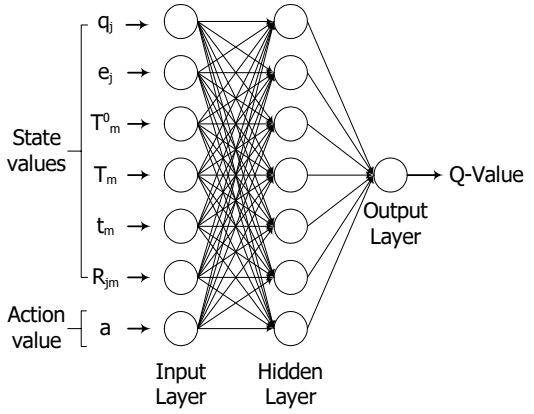
4.5 하이브리드 Q-학습 알고리즘

앞서 3.2절에서 언급하였듯이, 본 연구에서는 Q-학습과 ANN을 결합한 하이브리드 알고리즘을 제안한다. 본 연구에서 구성한 ANN은 피드포워드 역전파 뉴럴 네트웍(feed-forward back-propagation neural network, 이하 FBNN)이다. 본 FBNN은 입력층(input layer)에 현재의 상태와 그 상태에서 실행한 행동의 값을 입력하면 은닉층(hidden layer)을 거쳐서 출력층(output layer)으로 해당 Q-값을 산출해준다(그림 4 참조). 입력층으로 입력되는 값은 다음과 같이 현재의 상태와 그 때의 행동이다.

- 입력값 (상태):

① 현재 작업별 Queue에서 대기 여부(q_j, $1 \le j \le N$),
② 작업별 납기대비 tightness (e_j, $1 \le j \le N$),
③ 기계별 최근 가공 완료한 작업 ID (T_m^0, $1 \le m \le M$),
④ 기계별 현재 가공 중인 작업 ID (T_m, $1 \le m \le M$),
⑤ 기계별 최종 셋업 후 경과시간 (t_m, $1 \le m \le M$),
⑥ 작업과 기계간 rework 발생회수 (R_{jm}, $1 \le j \le N$, $1 \le m \le M$),

- 입력값 (행동):
⑦ 유휴 기계 ID ($1 \le m \le M$)와 할당할 작업 ID ($1 \le j \le N$)



[그림 3] Q-값 산출을 위한 FBNN

따라서 위와 같은 입력층의 표현방법을 따른다면, FBNN의 입력층에 필요한 노드의 수는 위의 상태 및 행동 입력값의 순서대로 $(N + N + M + M + M + NM) + (M + N) = 3N + 4M + NM$ 개다. 예를 들어 3대의 기계와 30개의 작업이 존재하는 공정은 192개의 입력층 노드가 필요하게 된다.

본 연구에서는 FBNN의 은닉층과 출력층을 위한 변환함수로 각각 TANSIG와 LOGSIG를 사용하였고, 이를 MATLAB 소프트웨어(nntool)를 사용하여 구현하였다. 이 외의 FBNN을 포함한 ANN에 대한 일반적인 설명은 생략하기로 한다.

적응형 스케줄링 문제를 위한 하이브리드 Q-학습 알고리즘을 위해, 기존 Q-학습의 틀 위에 상태와 행동에 따른 Q-값을 산출해 주는 'Q-FBNN'이라는 ANN 모듈을 삽입하였다. Q-FBNN과 Q-학습을 활용한 하이브리드 알고리즘의 전체적인 수행과정은 다음 ①~③의 절차를 반복 수행하는 것으로 이루어진다.

① 하이브리드 Q-학습 에이전트는 환경으로부터 전달받은 현재 상태를 인식하고, 현 상태에서 취하는 행동과 보상값을 이용하여 Q-FBNN의 학습을 수행한다.
② 에이전트는 Q-FBNN으로부터 Q-값을 산출하고 현 상태에서 CRR 규칙에 기반한 행동을 취한다.
③ 에이전트가 취한 행동은 환경에 의해 4.4절에서 정의한 보상함수에 따라 보상값을 전달받게 되고, 에이전트는 해당 보상값을 활용하여 Q-FBNN의 학습을 수행한다.

5. 실험결과 및 분석

5.1 비교대안과 실험 계획

비안정적인 rework 확률을 고려한 스케줄링 알고리즘에 관한 기존 연구를 찾아 볼 수 없기 때문에, 본 논문에서는 제안한 알고리즘의 성능평가를 위해 비교 대안 알고리즘으로 기존에 많이 사용되고 있는 할당 규칙들을 이용하기로 한다. 가장 대표적인 납기관련 할당 규칙인 EDD (Earliest Due-Dates)와 Lee *et al.*[4]가 제안한 ATCS (Apparent Tardiness Cost with Setups), 그리고 4.3절에서 소개된 CRR 규칙이 그것이다.

실험에 사용된 데이터의 생성 기준은 다음과 같다. 가공 시간과 작업 준비시간은 U[150,200]의 범위를 갖는 균일분포로부터, 작업투입시점은 0으로부터 평균 최대완료기간 (Expected Makespan : EM)의 R배까지인 [0, R*EM]의 범위를 갖는 균일분포로부터 생성하였다. 여기서 EM은 평균작업 준비시간과 평균가공시간의 합을 작업수 N과 곱한 후 기계수 M으로 나누어준 값이며, R은 작업투입시점의 범위모수로서 본 실험에서는 고정값으로 1.0이 사용되었다. 납기는 다음 식에 의해 생성되며, 수식에 사용된 값은 [0, 4]의 범위를 갖는 균일분포로부터 발생된다.

$$d_i = r_i + 2\alpha p_i$$

[표 1] 비안정적인 rework 확률

Types \ MC	1	2	3	4	5
A	NS	S	S	S	S
B	S	NS	S	S	S
C	S	S	NS	S	S
D	S	S	S	NS	S
E	S	S	S	S	NS
F	NS	S	S	S	S
G	S	NS	S	S	S
H	S	S	NS	S	S
I	S	S	S	NS	S
J	S	S	S	S	NS

[표 2] 실험계획

	Values	Total
No. of Jobs	100, 300, 500	3
No. of Job Types	5,10	2
No. of Machines	3,5	2
Combination		12
Problems per Combination		20
Total Problems		240

본 연구의 핵심 요소인 비안정적인 rework확률을 실험에 반영하기 위하여 [표 1]과 같이 특정 제품군에 속하는 작업타입과 특정 기계간의 rework 발생확률을 균일분포로부터 생성하였다. 표 1에서 'S'는 안정적인(stable) 상태로서 rework이 U[0, 0.001]로부터 발생되는 값(확률 값)에 의해 발생한다는 것을 의미하고, 'NS'는 비안정적인(non-stationary) 상태로 rework이 U[0.1, 0.4]에 따라 발생한다는 것을 의미한다. 본 실험에서 사용하는 벤치마킹데이터는 [표 2]와 같이 작업수와 제품타입, 그리고 기계수 등 변동될 문제요소의 조합으로 생성된 240개의 문제로 구성되어 있다.

그리고 사전 실험을 통해 학습률 모수 γ는 0.7로 결정하여 사용하였다. 본 연구에서 제시한 알고리즘은 MATLAB과 Visual Basic을 연동하여 구현하였고, 펜티엄4 2.4 GHz 컴퓨터에서 실험하였다.

[표 3] 알고리즘 성능 비교결과 (# of episodes = 300)

No of MCs	No of jobs	No of job types	EDD	ATCS	CRR	Proposed
3	100	5	10.49[a] (3.67)[b]	5.69 (2.87)	3.77 (1.35)	1.00
		10	9.57 (3.63)	5.19 (2.78)	4.03 (1.33)	1.00
	300	5	12.09 (3.49	6.55 (2.74)	3.9 (1.29)	1.00
		10	10.72 (3.57)	5.81 (2.77)	3.96 (1.33)	1.00
	500	5	12.49 (3.81)	6.77 (2.95)	3.77 (1.32)	1.00
		10	11.25 (3.79)	6.1 (2.94)	3.84 (1.33)	1.00
5	100	5	12.52 (4.26)	6.79 (3.41)	3.87 (1.29)	1.00
		10	11.75 (4.15)	6.37 (3.25)	3.85 (1.32)	1.00
	300	5	10.99 (4.44	5.96 (3.53)	3.57 (1.3)	1.00
		10	9.31 (4.12)	5.05 (3.4)	4.0 (1.31)	1.00
	500	5	12.86 (4.26)	6.97 (3.23	3.9 (1.36)	1.00
		10	11.52 (3.91)	6.24 (3.06)	3.98 (1.31)	1.00

a: Average of total tardiness

b: Standard error of the mean

5.2 실험 결과 및 분석

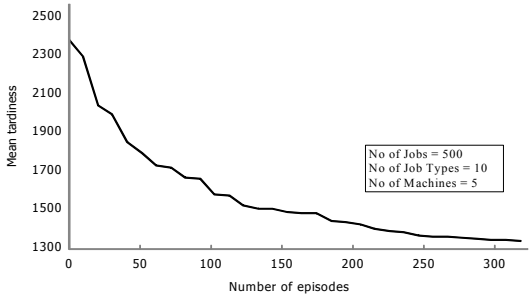
이 절에서는 표 2에서 제시한 240개의 실험 데이터를 사용하여 본 논문이 제시한 하이브리드 Q-학습 알고리즘의 성능을 목적함수인 평균지연시간(MT)의 최소화에 대해서 세 가지 대안 알고리즘(EDD, ATCS, CRR)과 각각 비교, 분석해 보기로 한다.

해의 질을 가늠하는 척도로서 본 실험에서는 3 가지 대안 알고리즘으로 실험하여 얻은 값을 하이브리드 Q-학습 알고리즘으로 얻은 값으로 나누어 준 비교값(comparison value)을 사용한다. 비교값은 다음의 식을 이용하여 구한다.

$$\text{비교값} = \frac{Average\ MT of\ Alternative\ Algorithms}{Average\ MT of\ Proposed\ Algorhim}$$

표 3은 240(12×20)회의 실험결과를 정리한 것이고, 이때 # of episodes의 값을 300으로 사용하였다. 표의 각 셀에 문제와 알고리즘에 따라서 20회씩 반복된 실험의 평균값과 표준편차 값을 표현하였다.

표 3의 결과에서 보듯이 제시한 알고리즘인 하이브리드 Q-학습 알고리즘이 대안 알고리즘인 EDD, ATCS, CRR보다 모든 경우에서 좋은 해를 제공하는 것을 볼 수 있다. 특히, rework확률을 전혀 고려하지 못하는 EDD, ATCS 보다는 월등히 좋은 결과를 도출해내고 있고, rework이 발생한 작업에 높은 우선순위를 부여하는 CRR보다도 비안정적인 rework확률을 고려하는 본 논문의 알고리즘이 뛰어난 성능을 보여주는 것을 알 수 있다. 그림 7은 에피소드 수의 증가에 따른 목적함수 값의 추이를 보여주고 있고, 하이브리드 Q-학습 알고리즘을 학습시키는데 필요한 전체 에피소드 수가 약 (250, 300)의 범위의 값을 갖는 것이 적정하다는 것을 보여주고 있다.



[그림 4] 에피소드 수에 따른 학습곡선

6. 결론

본 논문에서는 비안정적인 rework확률이 존재하는 제

조공정을 위한 적응형 스케줄링 알고리즘인 하이브리드 Q-학습 알고리즘을 제안하였다. Q-학습을 본 연구와 같은 제조공정에 적용하기 위해 상태변수, 보상함수 및 행동들에 대해 새로운 정의를 내렸고, 현실적인 문제로 확장이 용이하도록 Q-값을 학습시킬 목적으로 인공신경망을 도입하였다. 실험결과 하이브리드 Q-학습 알고리즘은 rework이 존재하고, 또한 그 rework 발생확률이 비안정적인 제조공정에서 아주 뛰어난 성능을 보인다는 것을 입증하였다. 따라서 본 알고리즘은 공정이 안정화된 경우는 물론이고, 고가의 장비가 처음 도입된 불안정한 상태의 제조공정에서도 그 효용성이 크다고 할 수 있다. 추후연구로는 제조공정의 불확실성에 효과적으로 대응하기 위하여 rework 판정과 연계된 스케줄링 알고리즘에 대한 연구가 필요하다.

감사의 글

Hankuk Academy of Foreign Studies 에 재학중의 이재우군의 노력과 기여에 감사드립니다.(The authors are grateful to Mr. Jae Woo Lee for his valuable suggestions.)

참고문헌

[1] Flapper, S.D.P., J.C. Fransoo, R.A.C.M. Broekmeulen and K. Inderfurth, 2002. Planning and control of rework in the process industries : review, Production Planning & Control, 13(1), 26-34.

[2] Kang, Y.H., Kim, S.S. and Shin H.J. 2010. A dispatching algorithm for parallel machines with rework processes, Journal of the Operational Research Society, 61, 144-155.

[3] Kuhl, M. E. and Laubisch, G. R. 2004. A simulation study of dispatching rules and rework strategies in semiconductor manufacturing, IEEE/SEMI Advanced Semiconductor Manufacturing Conference.

[4] Lee, Y.H. and Pinedo, M. 1997. Scheduling jobs on parallel machines with sequence-dependent setup times, European Journal of Operational Research, 100(3), 464-474.

[5] Lim, J.M., Bae, S.M. and Suh, J.J., 2006. A Learning based Algorithm for Traveling Salesman Problem, Journal of the Korean Institute of Industrial Engineers, 32(1), 61-73.

[6] Liu, J. J. and Ping, Y., 1996. Optimal lot-sizing in an imperfect production system with homogeneous reworkable jobs, European Journal of Operational Research, 91(3), 517-527.

[7] Sha, D. Y., Hsu, S. Y., Che, Z. H., and Chen, C. H., 2006. A dispatching rule for photolithography scheduling with an online rework strategy, Computers & Industrial Engineering, 50, 233-247.

[8] Teunter, R. H. and Flapper, S. D. P., 2003. Lot-sizing for a single-product production system with rework of perishable production defectives, OR Spectrum, 25, 85-96 .

[9] Zargar, A. M., 1995. Effect Of Rework Strategies On Cycle Time, Computers & Industrial Engineering, 29(1/4), 239-243.

신 현 준(Hyun Joon Shin) **[종신회원]**

- 1995년 2월 : 고려대학교 산업공학과(공학사)
- 1997년 2월 : 고려대학교 산업공학과(공학석사)
- 2002년 2월 : 고려대학교 산업공학과(공학박사)
- 2002년 5월 ~ 2004년 4월 : 미국Texas A&M대학교 연구원
- 2004년 6월 ~ 2005년 2월 : ㈜삼성전자 책임연구원
- 005년 3월 ~ 현재 : 상명대학교 경영공학과 조교수

<관심분야>
생산관리, 공급사슬망관리, 스케줄링, 금융공학

유 재 필(Jae Pil Ru) **[정회원]**

- 2009년 2월 : 상명대학교 산업정보시스템공학과(공학사)
- 2009년 3월 ~ 현재 : 상명대학교 경영공학과 석서과정

<관심분야>
금융공학, 생산관리