

Rework가 존재하는 이종병렬기계에서의 일정계획 수립

강용하¹ · 김성식¹ · 박종혁¹ · 신현준^{2*}

¹고려대학교 정보경영공학부 / ²상명대학교 산업정보시스템공학과

Scheduling Algorithm for Nonidentical Parallel Machines Problem with Rework

Yong Ha Kang¹ · Sung Shick Kim¹ · Jong Hyuck Park¹ · Hyun Joon Shin²

¹Division of Information Management Engineering, Korea University, Seoul 136-701

²Department of Industrial Information and Systems Engineering, Sangmyung University, Cheonan 330-720

This paper presents a dispatching algorithm for nonidentical parallel machines problem considering rework, sequence dependent setup times and release times. For each pair of a machine and a job type, rework probability of each job on a machine can be known through historical data acquisition. The heuristic scheduling scheme named by EDDR (Earliest Due Date with Rework probability) algorithm is proposed in this paper making use of the rework probability. The proposed dispatching algorithm is measured by two objective function value: 1) total tardiness and 2) the number of reworked jobs, respectively. The extensive computational results show that the proposed algorithm gives very efficient schedules superior to the existing dispatching algorithms.

Keywords: Rework probability, Nonidentical Parallel Machines, Scheduling, Dispatching Algorithm

1. 연구의 배경 및 목적

반도체(Semiconductor)나 박막트랜지스터 액정표시장치(TFT-LCD: Thin Film Transistor Liquid Crystal Display)시장은 제품 수명 주기 상 성숙기에 자리 잡고 있다. 따라서 전 세계의 제조업체들은 생산성 향상 및 제조원가 절감을 목표로 치열한 경쟁을 벌이고 있다. 더욱이 이들은 장치산업으로서 시설에 대한 투자비용이 크기 때문에 설비 투자 여부를 결정할 때 위험 관리 측면에서 신중하게 판단해야 하는 어려움이 있다. 이렇듯 치열한 경쟁과 투자 위험성이 존재하는 환경에서 경쟁력을 갖추려면 제품의 품질도 향상시켜야 하며 동시에 고객의 요구 사항도 절대적으로 만족시켜야 한다(Kang, 2003).

고객의 요구사항을 종합해 보면 크게 두 부분으로 요약될 수 있는데 그 하나는 약속된 납기 만족이고 또 하나는 품질이

다. 만일 고객 주문에 대한 납기 지연이 자주 발생하게 된다면 기업은 고객의 신용을 잃게 될 뿐만 아니라 그에 따라 시장에서의 점유율도 상실하게 된다. 그러므로 고객의 납기 만족을 생산관리의 최우선 목표로 삼아야 하며 이것은 생산관리의 핵심성과지표(KPI: Key Performance Indicator)가 된다.

이와 함께 품질이 나빠지면 고객의 신뢰도가 떨어질 것이고 이는 수익 창출에 있어서 해가 될 것이다. 또한 가공한 제품의 품질이 일정 기준에 미치지 못하게 된다면 이는 불량판정을 받아서 폐기되거나 재가공 작업을 거쳐서 양품을 만들어야 한다. 이때 가공을 위해 발생하는 비용도 무시할 수 없는 요소이기 때문에 좋은 품질의 제품을 만들어야 한다.

반도체 공장에서 가공을 위해 투입되는 기계에 따라서 제품의 제조사양(Specification, 이하 Spec)은 다양하게 변할 수 있다. 즉 어떤 기계에서 가공되느냐에 따라서 때로는 원하는

본 과제(결과물)는 교육인적자원부, 산업자원부, 노동부의 출연금으로 수행한 최우수실험실지원사업의 연구결과입니다.

* 연락처 : 신현준, 330-720 충남 천안시 안서동 98-20 상명대학교 산업정보시스템공학과, Tel : 041-550-5374, Fax : 041-550-5185,

E-mail : hjshin@smu.ac.kr

2007년 03월 접수; 2007년 05월 게재 확정.

Spec안에 있는 제품이 나올 수도 있고 그렇지 않은 경우가 발생할 수도 있는 것이다. 제품에 요구 되는 특정 Spec을 맞추지 못할 경우 그 제품은 사용이 불가능하기 때문에 재작업(rework)을 통하여 재가공을 하거나 아니면 버려지게 된다. 이는 재작업에 따른 가공시간과 그에 따른 비용이 추가적으로 발생하는 것을 초래하기 때문에 생산 계획을 수립할 경우 최소한의 재작업이 발생하도록 충분히 고려를 해야 한다.

반도체 공정에서 각 단위 공정별 통제(Control)의 정밀성 여부는 통제 알고리즘에도 영향을 많이 받지만 알고리즘 적용 이전에 공정의 작업순서에서도 상당한 영향을 받는다. 날로 고집적화되는 FAB(Fabrication)공정에서는 기존에 많이 사용하고 있는 공정 납기만 중점적으로 고려하는 Dispatching 알고리즘으로는 한계가 있다. 따라서 작업과 기계가 갖는 고유의 제조 특성을 반영하여 정밀한 Control을 가능하게 하는 새로운 알고리즘의 개발과 이의 시스템화가 반드시 필요하다고 할 수 있다. 이를 위해서는 1) 설비별 작업의 재작업 확률을 고려하여 정밀한 Control을 가능하게 해주는 작업순서의 생성과 작업 투입 기계 결정, 2) 공정납기를 어기지 않는 작업순서 생성이 필요하며 이는 제어 알고리즘과 유기적으로 통합된 Dispatcher가 필요함을 의미한다.

본 논문에서 다룰 문제 상황은 다음과 같다. 다양한 종류의 제품 타입을 갖는 작업들이 기계에서 가공된다. 이 때 제품 타입 별로 작업이 어떤 기계에서 가공이 되느냐에 따라서 서로 다른 rework 확률이 존재하며 이는 기계 자체의 특성에 의한 것이다. 비록 가공시간은 기계마다 동일하지만 rework 확률이 기계마다 다르기 때문에 본 논문이 다루고 있는 문제를 이중 병렬기계 문제라고 정의할 수 있다. 만일 임의의 기계에서 가공을 한 작업이 품질검사에서 실패하여 rework관정을 받았을 경우, 양품이 생산될 때까지 재가공 절차를 거친다. 즉 rework이 발생하지 않을 때까지 작업이 기계군 중 하나에서 반복적으로 투입, 가공과정을 거치게 된다. <Figure 1>은 이와 같은 전체시스템을 설명하고 있다.

Rework 확률이 존재하는 이런 상황에서 기존의 납기위주의 dispatching 규칙을 적용한다면 효율적인 결과를 얻기 힘들 것이다. 따라서 본 논문에서는 rework 확률을 고려하여 EDDR(Earliest Due Date with Rework probability)로 명명한 새로운 dispatching 알고리즘을 개발하고자 한다. 목적함수로는 납기준수의 측면에서 작업들의 전체납기지연값(Total Tardiness, 이하 TT)과 품질측면에서 재가공되는 작업의 수(Number of Rework, 이하 $NofR$)를 최소화하는 것이다. EDDR의 성능을 평가하기 위해 위 두 개의 목적함수에 대해 다양한 실험데이터를 이용하여 EDD(Earliest Due Date), MS(Minimum Slack), ATCS(Apparent Tardiness Cost with Setups) 등의 dispatching 알고리즘과 비교 분석을 통해 그 우수성을 보인다.

본 논문은 다음과 같이 구성되어 있다. 제 2장에서는 본 논문과 관련한 기존연구들에 대한 내용들이 있으며 제 3장에서는 본 논문이 제안한 병렬기계 알고리즘에 대하여 기술되어

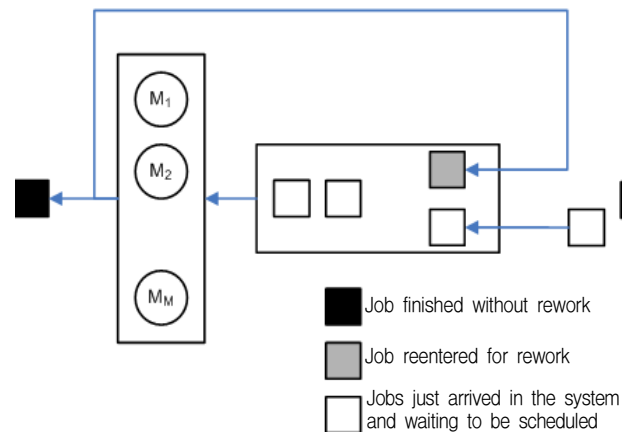


Figure 1. System configuration

있고 제 4장에서는 다양한 환경 하에서 실행된 실험 결과 및 분석을, 마지막으로 제 5장에서 결론 및 추후 연구 과제를 제시하는 것으로 마무리 한다.

2. 기존 연구

본 논문과 관련한 기존 연구는 크게 rework와 관련한 연구와 dispatching 규칙에 관한 연구로 나눌 수 있다.

Rework는 반도체 산업을 포함하여 많은 제조 산업에서 중요하게 고려되는 사항으로(Flapper 2002) 주로 비용의 최소화를 위해 최적의 생산배치크기나 재작업시기를 결정하는 문제 또는 최적의 rework 정책을 결정하는 것에 대한 연구가 진행되어 왔다.

Liu(1996)는 단일제품 생산 환경 하에서 단위시간당 평균이익을 최대화시키는 최적의 생산배치크기를 결정하는 방법을 제시하였고, Teunter *et al.*(2003)은 제품을 가공하다가 일정 수량(N)의 rework제품이 쌓이면 재가공을 시작하는 시스템에서 단위시간당 평균이익을 극대화시키는 N 의 크기를 정하는 문제에 관하여 연구하였다.

So와 Tang(1995)은 정상(Regular)제품과 재가공(rework)제품을 번갈아 가공하는 공정에서 Semi Markov Decision Process를 이용하여 평균운용비용을 최소화 시키는 최적의 운용정책(operating policy)을 결정하는 문제를 다루었고, Flapper와 Teunter(2004)는 논문에서 음식물과 같이 시간에 따라 품질이 나빠지는 제품들을 생산하는 공정에서 어느 시점에 얼마만큼의 rework 제품들을 가공해야 단위 시간당 평균이익이 높을지에 관하여 연구하였고 그에 따른 효율적인 rework 전략에 대하여 제시하였다.

Lee *et al.*(2005)는 반도체 MBT 공정에서 rework로 발생하는 제조원가 절감이익과 추가로 발생하는 재가공비용 및 재고비용 사이의 trade off를 고려한 rework 투입정책을 제안하였다.

Rework 정책을 dispatching rule과 결합한 연구도 있다.

Zargar(1995)는 'mother lot'과 'child lot'의 개념을 도입하고 그들을 처리하는 방법에 따라 4가지 rework 정책을 제시하였다. 또한 실험을 통해 rework 정책들이 cycle time에 미치는 영향에 관한 결과를 보여주었다. Kuhl and Laubisch(2004)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 있는 지에 관하여 시뮬레이션을 통해 비교 분석하였다.

Sha *et al.*(2006)는 반도체 포토공정 (photolithography)의 dispatching을 통해 전체 공정이 운용된다고 보고 고객의 납기를 만족시키고 동시에 높은 제품 품질을 얻기 위해서는 rework 전략과 dispatching 규칙을 적절하게 사용해야 한다고 말했다. 이때 세 가지의 rework 전략과 세 가지 dispatching 규칙의 조합으로 시뮬레이션을 수행하였다.

dispatching 규칙과 관련한 기존 연구로는 Demirkol과 Uzsoy (2000)의 연구가 있다. 그들은 근래에 들어 제품의 생산 방식과 과정이 복잡해지고 다양해지면서 제조업체는 생산능력을 효율적으로 사용하는데 어려움을 겪고 있으며 실제로 제조현장에서 나타나는 많은 일정계획 문제들은 지금까지 발표된 연구논문에서 언급한 것들보다 훨씬 난해하다고 지적했다.

순서의존적인 작업 준비시간과 작업투입시점을 고려하는 단일기계 또는 병렬기계에서의 일정 계획 수립에 대한 기존의 연구들을 살펴보면 다음과 같다.

Uzsoy *et al.*(1992, 1994)는 반도체 공정과 같은 고가의 설비를 운용하며, 순서의존적인 작업준비시간과 작업투입시점이 존재하는 제조공정은 일정계획을 수립할 때, 단순하고 직관적인 수작업 방식들을 사용한다면 생산성 및 비용측면에서 효과적인 결과를 기대하기가 힘들다고 밝히고 있다.

Kim and Bobrowski(1994)는 작업준비시간이 주문생산과 같은 생산시스템의 성능에 미치는 영향을 연구하기 위하여 시뮬레이션을 수행하였고 특히 작업준비시간이 순서의존적일 경우에는 스케줄링 문제를 풀 때 반드시 고려해야 하는 요소임을 보였다.

작업투입시점이 존재하는 제조환경에서는 기계의 유휴시간(idle time)을 최소화하기 위해 정확한 일정계획수립이 요구된다고 shin *et al.*(2002, 2003)은 밝히고 있다.

Lee *et al.*(1997 a)은 병렬기계에서 작업 순서에 따른 작업 준비 시간이 다른 경우에 대한 ATCS(Apparent Tardiness Cost with Setups)라는 알고리즘을 개발하였다. ATCS는 파라미터에 의해 그 성능이 좌우되는데 Park *et al.*(2000)은 파라미터를 실험에 의해 구하는 것이 아니라 신경망 이론(Neural Network)을 이용하여 파라미터를 구할 수 있도록 하였다.

Lee and Pinedo(1997 b)는 작업준비시점이 존재하는 병렬기계에서 납기 지연 가중치의 합을 최소화하는 문제를 풀기 위하여 할당규칙과 시뮬레이트 어닐링(Simulated Annealing : SA) 기법을 결합한 3단계의 발견적 기법을 제시하였다.

3. Scheduling Algorithm

3.1 문제 정의

각 제품 타입마다 어떤 기계에서 가공했느냐에 따라서 서로 다른 rework 확률이 존재한다면 작업들은 rework 확률이 작은 기계에서 가공을 하는 것이 납기 준수의 측면이나 rework 시 추가되는 비용 측면에서나 바람직할 것이다. 왜냐하면 rework 확률이 큰 기계에서 가공을 진행했을 경우 rework가 발생할 일이 그렇지 않은 경우보다 커질 것이고 이는 추가 가공을 위한 비용의 증가를 의미하기 때문이다. 하지만 그렇다고 모든 작업이 rework 확률이 작은 기계에서만 가공한다면 특정 기계에만 작업들이 몰리는 현상이 발생하게 되고 이는 rework에 따른 지연손실 및 품질비용은 줄이지만 자신의 차를 얻기 위하여 기계 앞에서 한없이 가공을 기다리는 상황이 발생하게 된다. 여기서 납기 만족과 rework 최소화 간의 상충관계(Trade-off)가 발생하므로 이 두 요인을 효과적으로 조절할 필요가 있다. 이 상충관계를 간단히 정리하면 다음과 같다.

- Rework가 발생한다면 최소한 그 작업의 셋업시간과 가공 시간만큼의 중복 처리 시간이 발생하게 된다. 그러므로 rework의 발생을 줄이는 것이 핵심 사항이라고 할 수 있다. 이를 위해서는 각 작업마다 rework 확률이 가장 낮은 기계로 들어가야 한다.
- 하지만 어느 한 type의 작업에 대한 주문이 많이 들어온 경우 그 작업들을 rework 확률이 작은 기계에서만 가공한다면 기계 앞에서의 대기 시간이 크게 증가하기 때문에 납기 지연시간이 많이 발생할 것이다. 이 경우에는 queue에 길게 늘어선 작업들을 다른 기계로 배분하여 작업을 처리하는 것이 좋을 것이다.

본 논문의 핵심은 위에 언급한 두 가지 사항을 절충하여 효율적인 생산계획을 수립하는 것이다. 다음 절에서는 예상완료 시간을 이용한 새로운 dispatching 알고리즘에 관하여 기술할 것이다.

3.2 EDDR 알고리즘

3.2.1 예상 완료시간

본 연구에서는 rework 가능성이 존재하는 상황에서 N 개의 작업을 $M(m=1, \dots, M)$ 개의 병렬기계로 스케줄 하고자 한다. N 개의 작업은 같은 제품타입(product type)을 갖는 C 개의 작업군으로 나뉘고, 이 때 작업군 J_c 를 제품타입이 c 인 작업들의 집합이라 한다. 작업군 J_c 에 속한 작업들의 인덱스를 $j(j=1, \dots, N_c)$ 라고 했을 때, 이들 작업 j 는 각각 납기(due date) d_j 와 작업투입시점(release time) r_j 를 갖는다. 각 작업 j 의 가공시간(processing time) p_j 는 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_c 에 속한 작업 i 의 가공을 마친 후 작업군 J_c 에 속한 작업 j 를 준비하는 데 소요되는 시간을 s_{cc} 라 할 때, $s_{cc} \neq s_{cc}$ 인 성질을 갖는 작업준비시간을 의미한다. 물론 작업 i 와 j 가 같은 제품군에 속한다면 $s_{cc} = 0$ 이다. 이 때 작업군 J_c 에 속한 작업 j 가 기계 m 에서 가공을 한 후 불량 판정을 받아서 재가공을 할 확률은 P_{cm} 이다.

t 라는 시점에 기계 m 이 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 j 는 기계 m 에서 가공을 완료한 후 다시 가공을 해야만 하는 상황이 발생할 수도 있다. 만일 작업 j 가 기계 m 에서 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 필요하게 된다. 이 시간을 R_j 라고 한다면 R_j 는 재작업 판정을 받은 작업이 기계 앞으로 돌아가 대기하다가 다시 기계에 투입될 때까지의 추정 체재시간(Estimated Sojourn Time)이라고 할 수 있다. rework가 발생하지 않아서 가공을 바로 마칠 수 있을 확률은 $(1 - P_{cm})$ 이고 rework가 발생하여 다시 기계 앞으로 돌아가 대기할 확률은 P_{cm} 이므로 결국 평균적으로 작업 j 를 기계 m 에서 가공할 경우 가공을 위해 필요한 시간은 $(1 - P_{cm})(s_{cc} + p_j) + P_{cm}(s_{cc} + p_j + R_j)$ 라고 할 수 있다. 이를 예상가공시간(Expected Processing Time, 이하 EPT)이라고 하자. 기계 m 에 작업 j 를 투입하는 시점이 t 이므로 작업 j 는 $t + (1 - P_{cm})(s_{cc} + p_j) + P_{cm}(s_{cc} + p_j + R_j)$ 에 가공을 완료 하게 되는데 이를 예상완료시간(Expected Completion Time, 이하 ECT)이라고 한다.

$$\begin{aligned} EPT &= (1 - P_{cm})(s_{cc} + p_j) + P_{cm}(s_{cc} + p_j + R_j) \\ &= (s_{cc} + p_j) + P_{cm}R_j \end{aligned} \quad (1)$$

$$\begin{aligned} ECT &= t + (1 - P_{cm})(s_{cc} + p_j) + P_{cm}(s_{cc} + p_j + R_j) \\ &= t + (s_{cc} + p_j) + P_{cm}R_j \end{aligned} \quad (2)$$

[Notation]

- $j(j = 1 \sim N)$: 작업 인덱스
- $c(c = 1 \sim C)$: 제품 타입 인덱스
- J_c : 제품타입이 c 인 작업들의 작업군 인덱스
- $m(m = 1 \sim M)$: 이종병렬기계 인덱스
- P_{cm} : 작업군이 J_c 인 작업 j 를 기계 m 에서 가공했을 때 재가공확률(rework probability)
- p_j : 작업 j 를 기계 m 에서 가공했을 때의 가공 시간(processing time)

s_{cc} : 제품타입이 c 인 작업 j 가 기계 m 에 투입되기 직전에 그 기계에서 가공을 완료한 작업의 제품타입이 c 일 때 작업 준비시간(setup time)

d_j : 작업 j 의 납기(due date)

r_j : 작업 j 의 투입 가능 시간(release time)

R_j : rework판정을 받은 작업 j 의 추정체재시간

\bar{s}_c : 제품타입이 c 인 작업 j 가 기계 m 에 투입되기 직전에 그 기계에서 가공을 완료한 임의의작업의 제품타입에 대한 평균작업 준비시간(average setup time)

재가공시 체재 시간인 R_j 는 rework시 한 번의 가공이 더 발생하게 되므로 R_j 는 작업을 가공 시 필요한 시간인 $\bar{s}_c + p_j$ 보다 같거나 큰 시간이라고 할 수 있다. 왜냐하면 작업 j 보다 우선순위가 높은 다른 작업들이 있다면 바로 기계에 투입되지 못하여 대기하는 시간이 발생할 수도 있기 때문이다. 그러므로 재가공하는데 소요되는 시간인 R_j 는 $NR \times (\bar{s}_c + p_j)$ 로 나타낼 수 있다. 여기서 NR 은 재가공시간 결정모수이고 이 값에 따른 EDDR의 성능은 제 4장에서 실험을 통하여 분석한다.

3.2.2 알고리즘 절차

본 논문에서 제시하는 dispatching 알고리즘은 EDDR(Earliest Due Date with Rework probability)이다. EDDR 알고리즘은 유휴(idle)상태인 기계가 발생할 경우 적용된다. 이 때 그 기계에서 가공했을 때 rework 확률이 가장 작은 작업타입인 작업들이 속한 작업군을 선호작업군이라고 하고 그렇지 않은 작업군을 비선호작업군이라고 한다. 마찬가지로 어떤 작업타입을 가공했을 때 rework될 가능성이 적은 기계를 선호기계라 하며, 그렇지 않은 기계를 비선호기계라 한다.

[EDDR 알고리즘 절차]

Step 1. 작업분류

1. 현재 queue에 있는 작업들을 작업타입별로 분류 기계가 유휴상태가 된 시점을 t 라 할 때 $r_j \leq t$ 인 작업 j 와 rework판정을 받아서 기계 앞으로 되돌아와서 대기 중인 작업 j 를 작업타입별로 분류한다.
2. 작업타입 별로 분류된 작업들을 납기가 빠른 순서(Earliest Due Date: EDD)로 정렬

Step 2. 대상 작업군 Ω 선정

1. 선호작업군
step 1에서 EDD로 정렬된 작업들 중 가장 선두에 있는 작업을 Ω 에 포함
2. 비선호작업군
모든 비선호작업군에 대하여 선호기계에서 먼저 투입된 작업의 가공이 종료되는 순간까지 기다린 후 가공을 했을 때의 예상 완료시간(ECT)(a)과 현재 유휴(idle)상태가 된 기계에서 가공했을 때의 예상 완료시간(ECT)(b)을 비교

하여 (a) > (b)인 작업들 중 가장 EDD인 작업을 선택하여 Ω 에 포함

Step 3. 대상 작업군에서 투입할 job 선택

1. Ω 에 포함된 작업들의 현재 유휴 상태인 기계에서의 예상 완료 시간(ECT)을 계산
2. 위 단계에서 계산한 예상 완료 시간이 가장 빠른 작업을 기계에 투입

step 1은 현재 스케줄 시점 t 에서 기계 앞에서 자신의 가공 순서를 기다리는 작업들(새로 주문된 작업들과 rework 판정을 받고 다시 기계 앞으로 돌아온 작업들)을 EDD로 정렬시킴으로써 현재 타입별로 가장 급한 작업이 무엇인지 알기 위한 단계이다.

step 2는 step 1에서 분류된 작업 순서에 의해 선호작업군과 비선호작업군에서 각각 후보 작업들을 선별하는 단계이다. rework가 되는 작업을 최소화하기 위해선 각 작업타입마다 자신이 가장 선호하는 즉, rework 확률이 가장 작은 기계에서 가공되는 것이 좋다. 하지만 만일 특정 제품 타입의 주문이 한꺼번에 몰려들어올 경우 이 원칙을 그대로 적용한다면 그 제품타입의 제품들은 대기 시간이 한없이 길어질 것이고 이는 심각한 납기지연 값의 상승을 불러오게 된다. 그러므로 현재 유휴 상태인 기계가 비선호기계인 작업군들은 비록 그것이 비선호 기계이기 때문에 재작업 확률이 높아질 지라도 어느 정도 작업들을 분산해서 가공할 필요가 있다. 일단 현재 유휴 상태인 기계가 비선호기계인 작업군에서는 바로 그 기계에 투입됐을 경우의 예상완료시간(b)과 작업 투입을 뒤로 미루더라도 선호 기계가 유휴상태가 될 때까지 대기하였다가 투입되어 가공했을 경우의 예상완료시간(a)을 비교한다. 또한 납기가 빠른 작업부터 비교를 시작하여 (a) > (b)인 작업이 나올 때까지 탐색한다. 이 때, (a) > (b)인 작업은 현재 유휴 상태인 기계에서 가공하는 것이 rework 발생 측면에서는 불리하지만 납기지연 단축 측면에서 이득이므로 상대적으로 높은 rework 확률을 감수하더라도 현재 유휴 상태인 기계에서의 가공을 허용하고 후보 작업으로 선정한다. (a) < (b)인 작업은 당장 가공을 할 수가 없어서 대기 시간이 발생하지만 대기를 하더라도 선호 기계에서 가공하는 것이 더 이득이라는 것을 말해 주기 때문에 굳이 비선호 기계에서 가공할 필요가 없는 작업이다. 비선호작업군에 속한 작업들 중 납기가 빠른 작업들부터 (a)와 (b)를 비교해 나가기 시작해서 (a) > (b)인 첫 번째 작업을 대상작업군 Ω 에 포함시킨다. step 3에서는 step 2에서 선별된 후보 작업들을 대상으로 실제 기계에 투입될 작업을 선정하는 단계이다. 대상작업군 Ω 에 포함된 작업들을 현재 유휴한 기계에서 가공했을 때 각 작업들의 ECT를 계산하여 그 중 가장 작은 ECT를 갖는 작업을 기계에 투입할 작업으로 최종 선정하고 유휴 기계에 투입한다. 즉, rework 확률을 고려하여 대상 작업의 완료 시간을 계산하여 가장 빨리 작업을 완료할 수 있는 작업을 기계에 투입하는 것이다.

Table 1. Rework probability

Type \ MC	1	2	3
A	0.1	0.4	0.3
B	0.3	0.2	0.4
C	0.2	0.3	0.1

3.2.3 수치 예제

본 절에서는 앞 절에서 언급한 알고리즘에 대한 수치예제를 제시한다. 병렬기계 수는 3대(M_1, M_2, M_3)이고 작업타입은 (A, B, C) 3가지이며, 작업군 $J_A = \{1, 2, 3, 4\}$ 이고 작업군 $J_B = \{5, 6, 7, 8\}$ 이며, 작업군 $J_C = \{9, 10, 11, 12\}$ 이다. 제품타입과 기계 별 rework 확률은 <Table 1>에 나와 있으며 순서의존적인 작업준비시간은 <Table 2>와 같다. <Table 1>은 기계 1의 경우 작업군 J_A 가, 기계 2의 경우 작업군 J_B 가, 기계 3은 작업군 J_C 가 선호작업군임을 설명해주고 있다.

Table 2. Setup time

Type \ Type	A	B	C
A	0	2	1
B	2	0	2
C	3	2	0

가공시간은 $p_2 = p_6 = p_{11} = 3, p_3 = p_7 = p_{10} = 4, p_4 = p_8 = p_{12} = 5$ 이다.

여기서는 재가공시간 결정모수 NR 을 1로 가정하여 문제를 푼다.

제품타입 A의 경우 $\bar{s}_{.A} = \frac{s_{AA} + s_{BA} + s_{CA}}{3} = \frac{5}{3}$ 이고, 제품타입 B의 경우 $\bar{s}_{.B} = \frac{s_{AB} + s_{BB} + s_{CB}}{3} = \frac{4}{3}$ 이고 제품타입 C의 경우 $\bar{s}_{.C} = \frac{s_{AC} + s_{BC} + s_{CC}}{3} = \frac{3}{3}$ 이다. 그러므로 $R_{j \in J_A} = \frac{5}{3} + p_{j \in J_A}, R_{j \in J_B} = \frac{4}{3} + p_{j \in J_B}, R_{j \in J_C} = 1 + p_{j \in J_C}$ 이다.

$t_1 (=2)$ 시점에 1번 기계는 가공 중이던 작업 1을 완료하여 유휴상태가 됐다고 가정해보자. 그 시점에 2번 기계에서는 작업 5, 3번 기계에서는 작업 9를 각각 가공 중이며 각각 가공 완료 시간은 $t_2 = 4, t_3 = 3$ 이다.

해 생성절차 중 1단계에 의해 작업들은 <Figure 2> (a)와 같이 제품타입별로 EDD순으로 정렬이 이루어지게 된다.

다음은 대상 작업군 Ω 를 선정하는 단계이다.

- 1) 먼저 선호작업군인 J_A 에서는 정렬된 작업 중 가장 선두에 있는 즉, 납기가 가장 빠른 작업인 작업 2가 Ω 에 포함된다.

2) 비선호작업군은 J_B 와 J_C 두 가지이다.

먼저 작업군 J_B 를 고려해보자. 알고리즘에 의해 스케줄을 기다리고 있는 작업들을 대상으로 선호기계인 2번 기계가 유힬해질 때까지 기다린 후 투입되는 경우(a)와 비선호기계인 1번 기계에 바로 투입되는 경우(b)의 ECT를 비교해야 한다.

대상 작업 중 가장 납기가 빠른 작업 6의 경우

$$(a) = t_2 + (s_{BB} + p_6) + P_{B2}R_6$$

$$= 4 + 0 + 3 + 0.2 * (\frac{4}{3} + 3) = 7.87 \text{ 이고}$$

$$(b) = t_1 + (s_{AB} + p_6) + P_{B1}R_6$$

$$= 2 + 2 + 3 + 0.3 * (\frac{4}{3} + 3) = 8.3 \text{ 이다.}$$

(a) < (b) 이므로 작업 6은 자신이 선호하는 기계인 M_2 에서 가공하는 것이 더 낫다.

다음 작업 7의 경우, (a)를 계산할 때 작업 7보다 납기가 더 빠른 작업 6이 선호기계의 가공이 끝나기를 기다렸다가 투입되는 것이 이득이라고 판명되었기 때문에 투입 가능 시점은 t_2 에다가 작업 6의 가공 시간을 더한 시점이 투입시점이 된다.

$$(a) = t_2 + (s_{BB} + p_6 + s_{BB} + p_7) + P_{B2}R_7$$

$$= 4 + 0 + 3 + 0 + 4 + 0.2 * (\frac{4}{3} + 4) = 12.07 \text{ 이고}$$

$$(b) = t_1 + (s_{AB} + p_7) + P_{B1}R_7$$

$$= 2 + 2 + 4 + 0.3 * (\frac{4}{3} + 4) = 9.07 \text{ 이다.}$$

(a) > (b)이므로 작업 7이 Ω 에 포함된다.

다음으로 작업군 J_C 를 고려해보자.

먼저 작업 10의 경우

$$(a) = t_3 + (s_{CC} + p_{10}) + P_{C3}R_{10}$$

$$= 3 + 0 + 3 + 0.1 * (1 + 3) = 6.4 \text{ 이고}$$

$$(b) = t_1 + (s_{AC} + p_{10}) + P_{C1}R_{10}$$

$$= 2 + 1 + 3 + 0.2 * (1 + 3) = 6.8 \text{ 이다.}$$

(a) < (b) 이므로 작업 10은 자신이 선호하는 기계인 M_3 에서 가공하는 것이 더 낫다.

다음 작업 11의 경우

(a)를 계산할 때 작업 11보다 납기가 더 빠른 작업 10이 선호기계의 가공이 끝나기를 기다렸다가 투입되는 것이 이득이라고 판명되었기 때문에 투입 가능 시점은 t_3 에다가 작업 10의 가공 시간을 더한 시점이 투입시점이 된다.

$$(a) = t_3 + (s_{CC} + p_{10} + s_{CC} + p_{11}) + P_{C3}R_{11}$$

$$= 3 + 0 + 4 + 0 + 3 + 0.1 * (1 + 3) = 10.4 \text{ 이고}$$

$$(b) = t_1 + (s_{AC} + p_{11}) + P_{C1}R_{11}$$

$$= 2 + 1 + 4 + 0.2 * (1 + 4) = 8 \text{ 이다.}$$

(a) > (b) 이므로 작업 11이 Ω 에 포함된다.

위의 과정을 통해 $\Omega = \{2, 7, 11\}$ 로 구성되며, <Figure 2> (b)는 이를 묘사하고 있다.

마지막으로 대상 작업군에서 투입할 작업을 선택하는 단계이다.

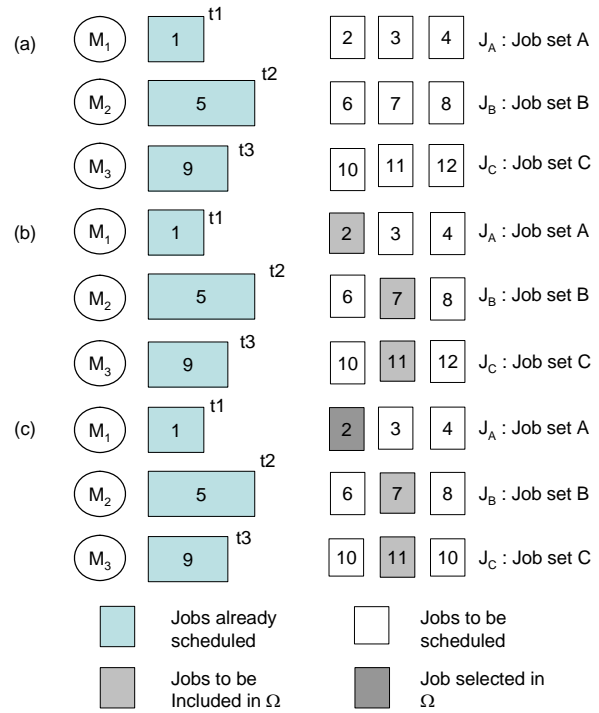


Figure 2. Working principles of EDDR : (a) sorting jobs in each job family by EDD in t_1 ; (b) choosing jobs to be included in Ω (c) Selecting the job having the minimum ECT value from Ω

현재 유힬 기계인 M_1 에서 가공했을 때 각 작업의 ECT는

i) ECT [작업 2]

$$= t_1 + (s_{AA} + p_2) + P_{A1}R_2$$

$$= 2 + 0 + 3 + 0.1 * (\frac{5}{3} + 3) = 5.47$$

ii) ECT [작업 7]

$$= t_1 + (s_{AB} + p_7) + P_{B1}R_7$$

$$= 2 + 2 + 4 + 0.3 * (\frac{4}{3} + 4) = 9.07$$

iii) ECT [작업 11]

$$= t_1 + (s_{AC} + p_{11}) + P_{C1}R_{11}$$

$$= 2 + 1 + 4 + 0.2 * (1 + 4) = 8 \text{ 이다.}$$

이 때, $\min(\text{ECT}[\text{작업 2}], \text{ECT}[\text{작업 7}], \text{ECT}[\text{작업 11}])$ 인 작업 2를 유힬기계인 M_1 에 투입할 작업으로 선택한다. 이와 같은 방식으로 대기 작업이 존재하지 않을 때까지 계속 반복 수행한다.

4. 결과 분석 및 결론

4.1 비교 대안과 실험 설계

rework 확률을 고려한 dispatching rule에 관한 연구는 아직 존

제하지 않기 때문에 본 논문에서 제안한 알고리즘의 성능평가를 위한 대안 알고리즘으로 기존에 많이 사용되고 있는 할당 알고리즘들인 MS, EDD, ATCS 규칙과 비교하였다.

본 연구에서 제시한 알고리즘의 효과를 검증하기 위하여 동일한 실험 환경 상태에서 모의실험을 통해 제시된 대안과의 비교를 실시하였다. 알고리즘의 성능을 비교하기 위한 척도로는 작업의 납기 지연 합(total tardiness), 재가공된 작업의 수(number of rework jobs)가 사용된다. 실험을 통해 나온 결과 값으로 대안 알고리즘과 성능을 비교하였다.

실험에 사용된 데이터의 생성 기준은 다음과 같다. 가공 시간과 작업 준비시간은 $U[150,200]$ 의 범위를 갖는 균일분포(uniform distribution)로부터, 작업투입시점은 0으로부터 평균최대완료기간(Expected makespan : T)의 R 배까지인 $[0, R \cdot T]$ 의 범위를 갖는 균일분포로부터 생성하였다. 여기서 T 는 평균작업 준비시간과 평균가공시간의 합을 작업수 N 과 곱한 후 기계수 M 으로 나누어준 값이며, R 은 작업투입시점의 범위 모수(Release Time Range Parameter)로서 본 실험에서는 (0.4, 1.0, 1.6)의 값이 사용되었다. R 의 값이 작을수록 작업이 빈번하게 도착하여 기계 앞에서 대기하는 작업수가 증가하게 되고 반대로 R 값이 커지게 되면 작업이 드문드문 도착하기 때문에 상대적으로 기계 앞에서 대기하는 작업의 수가 줄어들게 된다. 또한, R 값이 0이라면 해당 문제는 정적인 문제(Static Problem)이 되며 모든 작업이 스케줄 시점에 가용하다. 납기는 식 (3)에 의해 생성되며, 수식에 사용된 α 값은 $[-1, 4]$ 의 범위를 갖는 균일분포로부터 발생된다.

$$d_j = r_j + 2\alpha p_j \quad (3)$$

본 연구의 핵심 요소인 rework 확률은 <Table 4>에 정리된 것처럼 작업들의 기계선호도에 따라 확률값을 B(Best), N.B(Not bad), P(Poor)의 세 가지 단계로 나누었고 각각 $[0, 0.001]$, $[0.1, 0.2]$, $[0.2, 0.3]$ 의 범위를 갖는 균일분포로부터 생성하였다.

본 실험에서 사용하는 벤치마킹데이터는 <Table 3>과 같이 R 값과 작업수와 제품타입, 그리고 기계의 조합으로 생성된 720개의 문제로 구성되어 있다.

Table 3. Experimental design

	Values	Total
Parameters for Release Times(R)	0.4, 1.0, 1.6	3
Number of Jobs	100, 500, 1000, 2000	4
Number of Job Types	5, 10	2
Number of Machines	3, 5, 7	3
Combination		72
Problems per Combination		10
Total Problems		720

Table 4. rework probability

Type \ MC	1	2	3	4	5	6	7
A	B	P	N.B	N.B	N.B	N.B	N.B
B	N.B	B	P	N.B	N.B	N.B	N.B
C	P	N.B	B	N.B	N.B	N.B	N.B
D	N.B	N.B	N.B	B	P	N.B	N.B
E	N.B	N.B	N.B	P	B	N.B	N.B
F	N.B	N.B	N.B	N.B	N.B	B	P
G	N.B	N.B	N.B	N.B	N.B	P	B
H	B	N.B	N.B	N.B	P	N.B	N.B
I	P	N.B	N.B	N.B	B	N.B	N.B
J	N.B	B	N.B	N.B	N.B	P	N.B

본 연구에서 제시한 알고리즘은 모두 C#를 이용하여 구현하였고, 펜티엄4 2.4 GHz 컴퓨터에서 실험하였다

4.2 실험 결과 및 분석

이 장에서는 4.1에서 언급한 720개의 실험 데이터를 사용하여 본 논문이 제시한 EDDR 알고리즘의 성능을 두 가지 목적함수인 전체납기지연의 합(total tardiness: TT)과 재가공된 작업의 수(Number of reworked jobs: $NofR$)의 최소화에 대해서 세 가지 대안 알고리즘(EDD, MS, ATCS)과 각각 비교해 보기로 한다.

4.2.1 EDDR의 성능 분석

• 전체 납기지연의 합 (TT)

<Table 5>는 목적함수가 전체 납기지연 값의 합의 최소화일 때 4.1에서 언급한 720(72×10)회의 실험 중 $R = 0.4$ 일 때의 결과를 정리한 것이다. table의 각 셀에 문제와 알고리즘에 따라서 10회씩 반복된 실험의 평균값과 표준편차 값을 표현하였다.

EDDR 규칙의 성능을 대안알고리즘들과 비교하기 위하여 기계수, 작업수, 제품타입수 그리고 작업투입시점모수(R)를 기초로 하여 평가하였다. <Figure 3> (a), (b), (c), (d)은 각각 기계수, 작업수, 제품타입수, 작업투입시점범위모수(R)의 변화에 따른 전체납기지연시간의 합(Total tardiness)의 변화에 대하여 나타내고 있으며 해의 질을 가늠하는 척도로서 그림에서는 3 가지 대안 알고리즘과 제시된 알고리즘으로 실험하여 얻은 TT 값을 본 논문이 제시한 알고리즘인 EDDR로 얻은 TT 값으로 나누어 준 비교값(Comparison value)을 사용한다. 비교값은 식 (4)를 이용하여 구한다.

$$Comparison\ value = \frac{Average\ TT\ of\ Alternative\ Heuristics}{Average\ TT\ of\ EDDR} \quad (4)$$

Table 5. Performance results(TT) in case of R = 0.4

No of MC	No of jobs	No of product types	EDD	MS	ATCS	EDDR		
3	100	5	435,354 ^a (21,332) ^b	466,452 (27,208)	244,488 (29,887)	206,058 (21,902)		
		10	482,742 (20,799)	540,772 (24,242)	340,428 (12,126)	261,952 (29,333)		
		500	5	11,606,118 (248,624)	12,757,083 (516,159)	6,068,945 (351,392)	4,891,410 (153,289)	
			10	12,034,380 (229,986)	13,137,887 (465,428)	7,199,761 (262,120)	5,679,512 (155,850)	
	1,000	5	45,686,689 (742,918)	50,526,223 (1,057,118)	22,234,816 (828,899)	18,751,675 (293,230)		
		10	49,310,426 (667,953)	54,158,814 (595,800)	27,122,087 (358,778)	22,308,209 (606,136)		
		2,000	5	184,769,815 (2,011,533)	201,244,723 (2,267,240)	91,474,427 (3,423,245)	74,471,009 (1,134,948)	
			10	196,744,771 (1,958,111)	215,006,798 (3,563,315)	103,500,976 (2,361,002)	84,763,570 (2,046,351)	
	5	100	5	274,586 (12,249)	298,823 (21,368)	138,473 (19,358)	125,969 (19,740)	
			10	268,297 (13,762)	300,241 (21,527)	191,843 (10,667)	149,408 (12,467)	
			500	5	6,665,134 (141,259)	7,393,417 (269,033)	3,304,795 (188,783)	2,664,818 (84,530)
				10	7,219,593 (183,991)	7,907,072 (191,599)	4,065,145 (116,575)	3,181,906 (103,726)
1,000		5	27,174,253 (447,799)	29,436,054 (529,883)	12,989,554 (955,122)	10,581,456 (280,668)		
		10	29,034,908 (319,121)	32,125,571 (586,783)	15,366,696 (120,745)	11,838,928 (346,308)		
		2,000	5	110,769,955 (823,064)	120,522,545 (1,935,504)	53,811,302 (4,574,077)	42,555,009 (461,382)	
			10	118,899,483 (959,933)	128,765,040 (1,923,089)	60,069,723 (1,396,867)	46,634,184 (864,503)	
7		100	5	181,937 (11,306)	196,142 (12,637)	93,765 (7,657)	102,497 (9,548)	
			10	182,397 (8,989)	207,507 (12,790)	120,338 (8,274)	94,312 (11,083)	
			500	5	4,855,959 (91,245)	5,242,013 (232,829)	2,158,983 (131,013)	2,090,783 (107,115)
				10	4,978,544 (114,038)	5,559,172 (152,482)	2,687,394 (89,586)	2,116,627 (75,935)
	1,000	5	19,651,179 (218,246)	21,044,046 (359,827)	9,418,447 (486,734)	8,228,826 (192,931)		
		10	20,842,884 (254,670)	22,676,523 (442,790)	10,560,576 (251,545)	8,263,033 (279,104)		
		2,000	5	79,044,126 (942,455)	85,374,488 (1,007,316)	36,145,737 (2,385,784)	32,149,927 (471,155)	
			10	83,295,081 (731,928)	91,357,874 (1,270,085)	40,316,220 (1,222,479)	31,984,191 (707,382)	

^a : Average of total tardiness
^b : Standard error of the mean

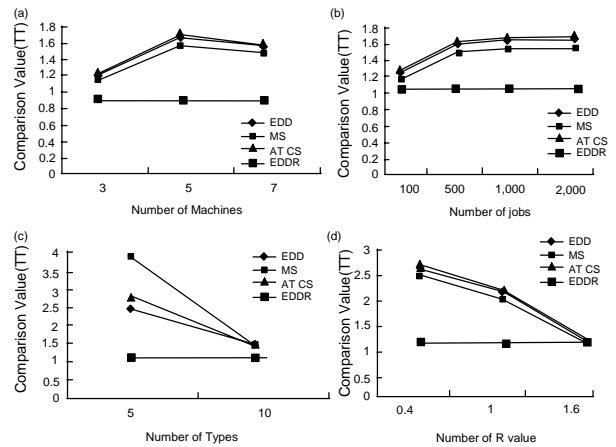


Figure 3. Performance comparison of EDDR with the comparative rules in terms of TT

<Figure 3>에서 보듯이 제시한 알고리즘인 EDDR 규칙이 대안 알고리즘인 EDD, MS, ATCS보다 모든 경우에서 좋은 해를 갖는 것을 볼 수 있다. 특히 기계수, 작업수, 제품타입수, 작업 투입시점범위모수(R)가 증가함에 따라 EDDR 알고리즘이 대안 알고리즘보다 더욱 좋은 결과를 도출해내고 있다. 전체 납기지연의 합의 관점에서는 모든 문제에서 EDDR > ATCS > EDD > MS의 순으로 좋은 성능을 보였다.

• 재가공된 작업의 수 (NofR)

<Table 6>은 목적함수가 NofR의 최소화일 때 4.1에서 언급한 720(72×10)회의 실험 중 R = 0.4일 때의 결과를 정리한 것이다. TT와 마찬가지로 table에 나와 있는 각 셀에 문제와 알고리즘에 따라서 10회씩 반복된 실험의 평균값과 표준편차 값을 표현하였다.

EDDR 규칙의 성능을 대안 알고리즘들과 비교하기 위하여 기계수, 작업수, 제품타입수 그리고 작업 투입시점범위모수(R)를 기초로 하여 평가하였다.

<Figure 4>. (a), (b), (c), (d)는 기계수, 작업수, 제품타입수, 작업 투입시점범위모수(R)에 따른 재가공된 작업의 수 (NofR)의 변화에 대하여 나타내고 있다.

<Figure 4>에서는 세 가지 대안 알고리즘과 제시된 알고리즘으로 실험하여 얻은 NofR값을 제시된 알고리즘인 EDDR로 얻은 NofR값으로 나누어 준 비교값을 해의 질을 가늠하는 척도로 사용한다.

$$Comparison\ value = \frac{Average\ NofR\ of\ alternative\ Heuristics}{Average\ NofR\ of\ EDDR} \quad (5)$$

그림에서 보듯이 모든 경우에 제시한 알고리즘이 대안 알고리즘인 EDD, MS, ATCS보다 좋은 해를 갖는 것을 볼 수 있다. 특히 기계수, 작업수, 제품타입수, 작업 투입시점범위모수(R)가 증가함에 따라 EDDR 알고리즘이 대안 알고리즘보다 더욱 좋은 결과를 도출해내고 있다.

Table 6. Performance results(*NofR*) in case of $R = 0.4$

No of MC	No of jobs	No of product types	EDD	MS	ATCS	EDDR		
3	100	5	21a (6)b	19 (3)	23 (10)	13 (3)		
		10	19	19	20	17		
	500	5	5	112 (16)	105 (12)	130 (20)	46 (8)	
			10	92 (9)	88 (12)	96 (15)	72 (11)	
		1,000	5	201 (18)	192 (12)	207 (34)	91 (9)	
			10	190 (16)	183 (13)	188 (16)	145 (26)	
	2,000	5	5	410 (22)	391 (19)	439 (53)	179 (15)	
			10	374 (24)	360 (24)	371 (41)	290 (39)	
		5	100	5	22 (5)	21 (5)	22 (9)	9 (6)
				10	20 (5)	17 (3)	19 (7)	16 (4)
	500	5	5	95 (11)	93 (11)	110 (20)	15 (6)	
			10	88 (16)	85 (9)	89 (13)	54 (11)	
1,000		5	200 (15)	187 (14)	213 (53)	20 (8)		
		10	183 (14)	177 (15)	190 (11)	90 (12)		
2,000	5	5	404 (25)	384 (22)	440 (136)	18 (5)		
		10	382 (22)	360 (22)	389 (41)	194 (20)		
	7	100	5	21 (3)	19 (4)	23 (6)	17 (5)	
			10	20 (5)	19 (4)	20 (5)	14 (4)	
500	5	5	96 (10)	90 (12)	82 (19)	35 (8)		
		10	91 (12)	83 (12)	92 (15)	43 (10)		
	1,000	5	194 (16)	181 (15)	197 (29)	61 (8)		
		10	177 (13)	170 (15)	190 (18)	67 (11)		
2,000	5	5	399 (17)	373 (16)	352 (88)	119 (19)		
		10	367 (23)	351 (19)	381 (65)	136 (16)		

^a : Average of number of rework

^b : Standard error of the mean

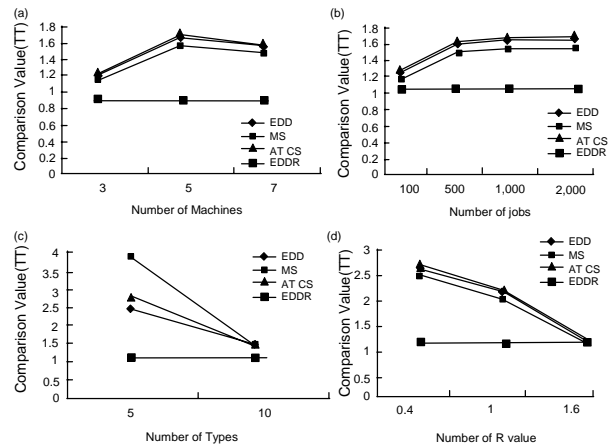


Figure 4. Performance comparison of EDDR with the comparative rules in terms of *NofR*

하지만 *TT*와는 다르게 *NofR*은 제품타입이 증가하거나 R 값이 증가하게 되면 대안알고리즘과의 차이가 상당히 줄어들게 되었다. 이는 제품타입이 늘어나게 되면 하나의 기계를 두고 여러 개의 선호작업군이 발생하게 되어 경쟁에서 진 작업군의 작업은 상대적으로 rework 확률이 큰 기계에 투입되게 되고 결국 이는 rework되는 작업의 수를 증가시키는 결과를 만들게 되기 때문이다. R 값의 증가가 *NofR*을 증가시키는 이유는 상대적으로 주문이 발생하는 간격이 늘어나게 되어서 어떤 기계가 유희하게 되었을 때 그 기계가 선호하는 작업군의 작업이 존재하지 않을 가능성이 있기 때문이다.

4.2.2 NR 에 따른 EDDR의 성능 분석

제 3장에서 언급된 NR 값이란 재가공을 하기로 결정했을 때 추가적으로 발생하는 가공시간을 결정하는 재가공시간 결정모수이다. 본 절에서는 재가공시간에 따라서 알고리즘의 성능이 어떻게 변하는 지에 관하여 실험을 바탕으로 설명할 것이다. 총 72개의 문제 중 R 값이 0.4, 작업수 1000, 기계수 3 그리고 작업타입은 10인 문제에서의 결과값을 출력하였고 NR 값은 1부터 20까지 0.5씩 증가시키면서 실험을 수행하였다.

<Figure 5> (a), (b)는 두 가지 목적함수에 대하여 NR 의 변화에 따른 목적함수 값의 변화에 대하여 보여주고 있다. 실험 결과 두 가지 목적함수 모두 NR 값이 작을 때 더 좋은 해가 나온다는 사실을 보여주고 있다.

최고 성능을 보인 NR 값은 *TT*와 *NofR*의 경우 모두에서 1이었다. 즉, 재가공 시간이 작을수록 EDDR의 성능이 좋다는 결론을 도출할 수 있는 것이다. 재가공 판정을 받은 작업은 이미 그전에 가공순서에서 높은 우선순위를 가졌기 때문에 재가공 판정을 받아서 다시 기계로 돌아가더라도 높은 우선순위를 거의 유지할 것이다. 즉, 재가공이 이루어지더라도 대부분 바로 다시 기계에 투입될 것이다. 이는 최고해를 갖는 NR 값이 작은 수라는 사실과 연관이 있다. 나머지 71개 문제 모두 대부분 위 문제와 유사한 패턴의 결과를 보여주었다.

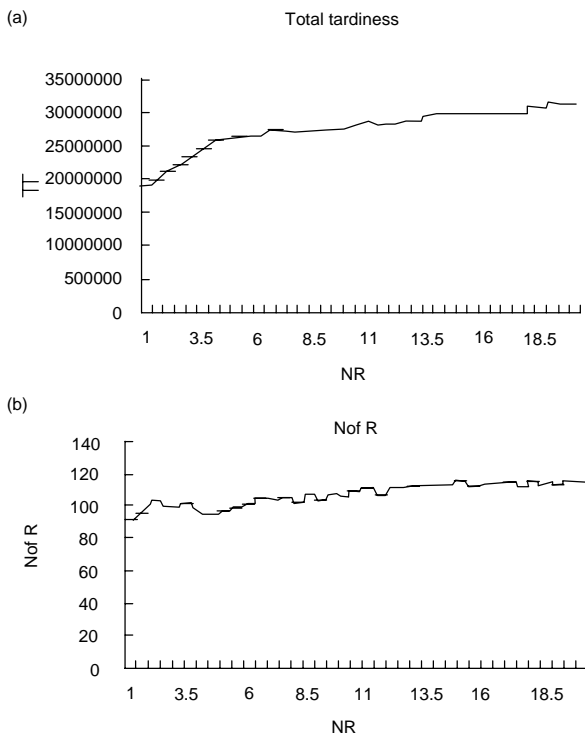


Figure 5. Performance comparison of EDDR according to the NR

5. 결론 및 추후연구

본 연구에서는 작업이 속한 제품타입과 가공되는 기계에 따라 rework 확률이 달라지는 이종병렬기계 문제에서 작업들의 전체납기지연값(Total Tardiness, 이하 TT)과 재가공되는 작업의 수(Number of Rework, 이하 $NofR$)를 최소화하는 EDDR 알고리즘을 제시하였다. 두 가지 목적함수에 대하여 각각 실험을 시행하였고 그 결과 다른 대안알고리즘인 EDD, MS, ATCS 보다 본 연구에서 제안한 EDDR 규칙이 좋은 해값을 산출하였다.

추후에는 작업이 재가공 판정을 받은 후 다시 기계에 투입되어 가공을 완료할 때까지 걸리는 시간을 정확하게 예측할 수 있는 알고리즘을 개발할 필요가 있다. 또 근시안적인 한계를 지니고 있는 EDDR 알고리즘에 의해 생성된 해를 향상시켜 최적값에 근사한 해를 구할 수 있는 방법에 관한 연구도 필요하다.

참고문헌

- Demirkov, E. and Uzsoy, R. (2000), Decomposition methods for re-entrant flow shops with sequence-dependent setup times, *Journal of Scheduling*, **3**, 155-177.
- Flapper, S. D. P., Fransoo, J. C., Broekmeulen, R. A. C. M., and Inderfurth, K. (2002), Planning and control of rework in the process industries : review, *Production Planning & Control*, **13**(1), 26-34.
- Flapper, S. D. P. and Teunter, R. H. (2004), Logistic planning of rework with deteriorating work in process, *International Journal of Production Economics*, **88**, 51-59.
- Kang, Y. H. (2003), A scheduling scheme for parallel machines where jobs demand repetitive processing, a thesis for a master degree, Dept. of I.E., Korea Univ.
- Kim, C. O. and Shin, H. J. (2003), Scheduling Jobs on Parallel Machines : A Restricted Tabu Search Approach, *International Journal of Advanced Manufacturing Technology*, **22**(3-4), 278-287.
- Kim, S. C. and Bobrowski, P. M. (1994), Impact on sequence-dependent setup time on job shop scheduling performance, *International Journal of Production Research*, **32**(7), 1503-1520.
- 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*.
- Lee, D. H., Ko, H. H., and Kim, S. S. (2005), A study of strategy for planning of rework in semiconductor monitoring burn in test process, *IE interfaces*, **18**(3), 350-360.
- Lee, Y. H., Bhaskran, K., and Pinedo, M. (1997), A heuristic to minimize the total weighted tardiness with sequence-dependent setups, *IIE Transactions*, **29**, 45-52.
- 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.
- 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.
- Park, Y. S., Kim, S. Y., and Lee, Y. H. (2000), Scheduling Jobs on Parallel Machines Applying Neural Network and Heuristic Rules, *Computers & Industrial Engineering*, **38**, 189-202.
- 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.
- Shin, H. J., Kim, C. O., and Kim, S. S. (2002), A tabu search for a single machine scheduling with release times, due dates and sequence-dependent setup times, *International Journal of Advanced Manufacturing Technology*, **19**, 859-866.
- So, K. C., Tang, C. S. (1995), Optimal operating policy for a bottleneck with random rework, *Management Science*, **41**(4), 620-636.
- 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.
- Uzsoy, R., Lee, C. Y. and Martin-Vega, L. A. (1992), A review of product planning and scheduling models in the semiconductor industry Part I : Systems characteristics, performance evaluation and production planning, *IIE Transaction on Scheduling and Logistics*, **24**(4), 47-60.
- Uzsoy, R., Lee, C. Y. and Martin-Vega, L. A. (1994), A review of product planning and scheduling models in the semiconductor industry Part II : Shop-floor control, *IIE Transaction on Scheduling and Logistics*, **26**(5), 44-55.
- Zargar, A. M. (1995), Effect Of Rework Strategies On Cycle Time, *Computers & Industrial Engineering*, **29**(1/4), 239-243.