

# 유연생산시스템에서의 실시간 혼합 일정계획 알고리즘

노인규 · 조철형

한양대학교 산업공학과

## A Real-time Mixed Scheduling Algorithm in Flexible Manufacturing System

In-Kyu Ro · Chull-Hyung Cho

Recently, the researches of real-time scheduling that manage flexibly system movement are being progressed by using simulation. But because the existing researches are focused on state change of whole system and selected dispatching with real-time, it could not reflect the state changes of each work center accurately. Therefore, the main core of this paper is to examine the state of workcenter, to select the optimal dispatching dynamically through simulation, and to present the algorithm that can manage actively on state of workcenter. Also, performance measure was used the mean tardiness and mean flow time that are used existing researches.

### 1. 서 론

#### 1.1 연구의 배경 및 목적

현대에 제조업이 직면하고 있는 어려움 가운데 가장 일반적인 것 중 몇 가지가 바로 제품수명의 단축화와 제조기술의 다양화 그리고 시장의 변화를 들 수 있다. 이러한 어려움이 점차 커져가는 과정에서 제조에 관련된 연구들은 여러 가지 새로운 생산이론을 개발하고 시대의 흐름에 따라서 그 이론을 지속적으로 발전시켜 왔다. 그 가운데 생산시스템의 효율성과 유연성의 향상이 중요 과제가 되었으며, 효율성이 뛰어난 흐름생산방식(Flow Shop)과 유연성이 뛰어난 주문생산방식(Job Shop)의 장점을 모두 갖추고 있는 유연생산시스템(Flexible Manufacturing System)에 관심이 집중되고 있다.

유연생산시스템은 중소량단위의 생산환경 내에서 대량생산효과에 접근하기 위하여 NC machine-tool과 Material Handling System, 그리고 그 밖의 외부 생산장비를 컴퓨터로 통합 컨트롤 하는 생산방식이다(Viswanadham and Narahari, 1992).

이러한 유연생산시스템의 장점은 다음과 같이 3가지로 분류할 수 있다(Ishii and Talavage, 1991).

- 생산시간(Lead Time)과 제공품(WIP)을 줄이고, 시장변화에 빠르게 대처한다
- 제품의 품질을 높인다
- 장비의이용률을 높인다

이와 같은 장점들은 유연생산시스템이 제조업자들에게 아주 중요한 관심사가 되어온 이유가 된다. 그러나 유연생산시스템은 그 복잡성으로 인해 설치와 운영에 많은 어려움이 있는데, 특히 부품 종류의 증가와 다양한 수행도의 필요성으로 일정계획(scheduling)은 더욱 복잡하고 어려운 문제가 되고 있다. 따라서, 복잡한 환경 내에서 최적의 일정계획을 수립하는 것은 유연생산시스템의 중요한 연구 분야로 자리 잡게 되었다.

제조업자는 경쟁력 있는 제품을 만들기 위하여 생산에 관련된 시간들에 중점을 두어 생산환경변화에 빠른 대처를 하여야 한다. 이것이 총괄생산계획(Aggregate Production Planning)이나 기준생산계획(Master Product Scheduling)과 같은 중장기 생산계획을 뒷받침해 주는 단기 생산계획의 연구가 중요시되는 이유이다.

이 연구에서는 이중에서 유연생산시스템의 일정계획문제를 다루며 유연생산시스템의 특징인 설비의 유연성을 이용해서 시스템의 변화에 실시간으로 대처하고 각 작업장의 상태를 반영하는 실시간 혼합일정계획 기법(Real-time Mixed Scheduling)을 제시한다. 그리고 제시한 알고리즘의 성능을 판단하기 위하여 평균흐름시간(Mean Flow Time)과 납기지연시간(Mean Tardiness)을 사용하였다. 이 수행척도를 사용한 이유는 첫째로 작업흐름시간의 최소화는 재공품감소와 이용률의 향상 등 부가적인 장점이 있기 때문에 아주 중요한 수행척도이다. 두 번째는 기존연구들이 자신의 우수성을 평균흐름시간과 납기지연시간으로 입증하였기 때문에 이 연구에서도 이 수행척도를 통하여 우수성을 입증한다. 또한, 이 연구의 모의실험은 실험 계획법에 의거하여 적당한 수행횟수를 계산하고 결과를 산출하였다.

## 1.2 기존연구 고찰

동적일정계획(Dynamic Scheduling)에 관한 알고리즘을 설계하기 위하여 가장 일반적인 접근방법은 우선 순위규칙이라고 불리는 할당규칙(Dispatching Rule)을 사용하는 것이다(Blackstone, et al., 1982). 일정계획시스템의 의사결정자는 그들의 시스템 형태 및 조건에 가장 적합한 할당규칙을 선택하는 데 있어서 아주 큰 어려움을 가지게 되며 이런 문제를 해결하기 위하여 많은 연구가 진행되어 왔다.

Blackston과 Phillips(1982)는 어떤 단일 할당규칙이 모든 작업 환경 안에서 다른 규칙들 보다도 언제나 수행도가 뛰어날 수 없다고 하였으며, 이는 변화 가능한 작업환경을 모두 고려할 수 있는 단일규칙은 개발될 수 없다는 것이다. 바로 이런 점 때문에 시스템의 상황변화에 민감한 동적할당 알고리즘이 연구되었으며, 특히 개별공정하에서 시스템의 상태 변화를 그대로 반영하는 동적 일정계획 알고리즘은 여러 가지 방법론을 통해서 연구되었다.

Kiran과 Smith(1984)는 모의실험에 의한 할당규칙 연구를 위한 최적조건들을 제시하였다. 예를 들어 할당규칙들에 관한 연구에서 가장 적당한 대기행렬 길이를 구하기 위해서는 기계 가동률을 0.85에서 0.95 사이로 설정하고 모의실험을 해야 한다고 제시하였다. 그러나 이 연구도 여러 가지 다양한 주문들이 작업조마다 다르게 배당되는 시스템에서 기계가동률을 일정하게 유지하는 것은 현실적으로 어렵기 때문에 동적시스템에서는 적용하기 부적합하다.

Thesen과 Lei(1986)는 실시간 일정계획 문제를 접근하기 위하여 전문가 시스템을 이용하였다. 실험을 통하여 시스템의 변화에 따라서 최적 할당규칙에 대한 정보를 데이터베이스화 시키고, 다시 같은 상황이 발생할 경우 전문가 시스템의 도움을 받아 실시간으로 할당규칙을 선택하는 방법론이다. 또 그들은 자신의 알고리즘을 통하여 일정 단위기간동안에 여러 가지 할당규칙을 조합해서 사용하는 것이 지속적으로 한 가지 할당규칙을 사용하는 것보다 더욱 좋은 수행도를 얻을 수 있다는 것을 보여주었다.

Wu와 Wysk(1989)는 여러 가지 할당규칙을 시스템환경에 따라서 적용하는 기법을 고안하였고, 이산사건 모의실험(Discrete Event Simulation)을 통하여 최소시간에 할당규칙을 선택하는 알고리즘을 개발하였다. 그들도 다양한 할당규칙의 조합에 대한 우수성을 입증하였다.

Ro와 Kim(1990)은 유연생산시스템의 운영 통제를 초기 투입 단계시 작업물 선정 문제, 일반 투입단계시 작업물 선정 문제, 작업물 기계할당 문제, 공정 선택 문제, 무인 운반차 작업분배 문제, 무인 운반차 이동경로 문제 등 6단계로 분류한 후 총소요 시간(Makespan), 생산흐름시간, 평균지연시간, 최대지연시간, 시스템 가동률을 모두 고려한 새로운 기계선택 규칙을 개발하였다.

Ishii와 Talvage(1991, 1996)는 일시상태(Transient-based State)에서 실시간으로 여러 할당규칙을 적용하는 알고리즘을 제시하였다. 이 연구에서 그들은 시스템 전체의 상태변화에 중점을

두고 납기를 고려한 알고리즘을 제시하였으나 상태변화를 측정하는 수리식이 평균값을 이용하고 있기 때문에 시스템은 각 작업장의 상태변화에 매우 둔감하게 된다.

Huajie와 Jian(1996)은 역전과 신경망을 원하는 목적에 가장 적합한 할당규칙을 동적으로 선택할 수 있도록 구성하였다. 이들은 신경망을 학습시킬 때, 작업순서와 작업시간을 입력데이터로 사용하였고 목표출력을 위해 이산사건 모의실험(Discrete Event Simulation)을 수행하였다.

Pierreval과 Mebaki(1997)는 제안된 시스템의 목적을 두 가지(제1목적과 제2목적)로 나누어 시스템 설정과 환경에 따라서 징후(Symptom)를 설정하여 표준규칙과 진단규칙을 정하고, 여기에 따라서 할당규칙을 동적으로 적용시키는 알고리즘(Shift From Standard Rules)을 제시하였다.

앞에서 살펴본 바와 같이 기존의 연구에서는 작업장의 상태 변화보다는 시스템 전체의 상태변화에만 중점을 두었기 때문에 작업장의 변화를 민감하게 고려할 수 없었다. 이를 위하여 이 연구에서는 각각의 작업장에 여러 할당규칙조합을 동적으로 적용시켜 정확한 작업장상태를 반영하는 실시간 혼합 일정 계획에 관한 알고리즘을 제시한다.

## 1.3 연구의 구성 및 내용

이 연구에서는 작업장의 상태를 점검하고 거기에 따른 할당 규칙들을 모의실험을 통하여 동적으로 선정해서 각 작업장의 상태변화에 능동적으로 대처할 수 있는 기법을 제시하여 납기 지연시간과 평균흐름시간을 최소화하는 데 그 목적이 있다.

제 2장에서는 이 연구에서 제시하는 알고리즘의 기초가 되는 실시간 일정계획과 혼합할당규칙에 관하여 알아본다. 제 3 장에서는 2장에서 설명한 실시간 일정계획과 혼합할당규칙을 이용하여 실시간 혼합할당규칙에 대한 알고리즘을 설명한다. 제 4장에서는 기존연구에서 제시한 단일 할당규칙과 실시간 일정계획 알고리즘, 그리고 이 연구에서 제시한 알고리즘을 비교하기 위한 시스템 환경을 설정하고 그것에 따른 모의실험 방법을 소개하며, 모의실험결과를 분석하여, 기존의 할당 알고리즘에 대한 본 연구의 우수성을 입증한다. 마지막으로 제 5 장에서는 결론 및 추후 연구과제를 제시한다.

## 2. 실시간 일정계획과 혼합 일정계획 알고리즘

유연생산시스템 특성상, 시스템 내로 여러 가지 종류의 부품들이 매 시간에 다양한 시스템 상태에서 들어온다. 그러나 기존에 일정계획을 비롯한 유연생산의 여러 분야에서 보여준 많은 연구들이 이와 같은 점을 모두 고려하지 못하였다. 그 후에 발표된 연구들은 바로 이런 점을 극복하기 위한 알고리즘을 제시하였으며, 일정계획 분야에서는 시스템 상태를 자동으로 점검하여 그 상태변화를 시스템에 반영하여 주는 동적 일정계획 기법에 관한 연구들이 나오면서 활성화되기 시작하였다.

동적 일정계획 기법을 이용한 많은 연구들이 우수한 발견적

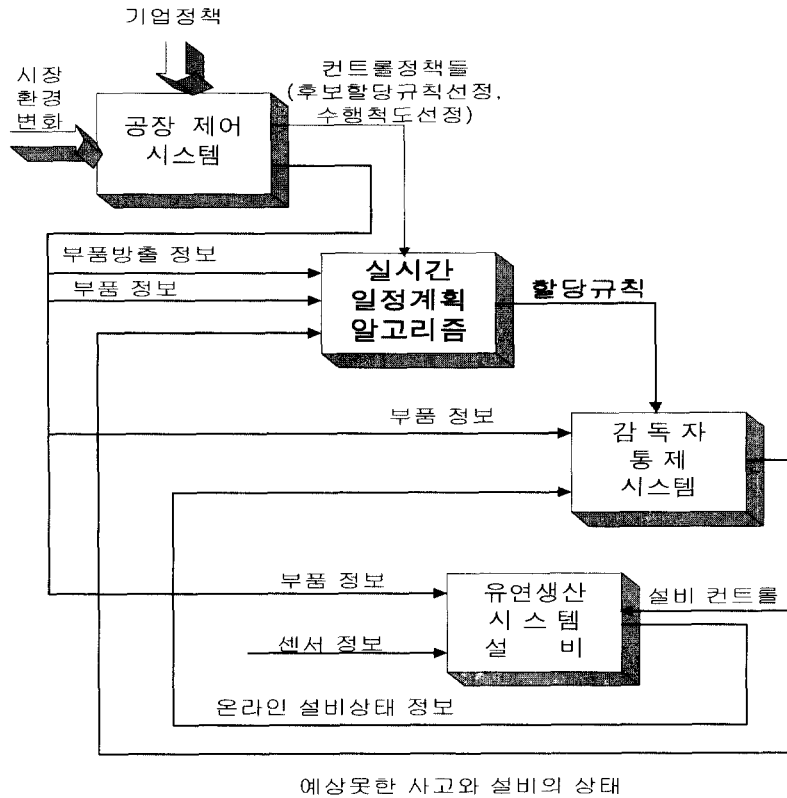


그림 1. 실시간 일정계획 알고리즘의 영역 및 역할.

해법들을 제시하여 왔는데, 그 중에서 가장 많이 이용하는 방법론으로는 시스템 상태 변화를 지식기반데이터베이스(Knowledge-based DB)를 가지고 전문가 시스템을 구현한 후 질의를 하여 시스템 환경을 변화시키는 기법과 본 연구에서 제시하는 시간변화에 따라서 시스템 상태를 점검하는 실시간 일정계획 기법이 있다. 그러나 이 기법들에 관한 연구들에서도 시스템 상태를 점검할 때 시스템 전체의 변화에만 중점을 두었고 하나의 할당규칙을 모든 작업장에 동시에 적용하기 때문에 정확한 작업장의 상태를 반영하지 못하는 단점이 존재한다. 순수하게 각각의 작업장의 상태를 점검하여 각 작업장마다 할당규칙을 달리 적용하기 위한 기법으로는 혼합일정계획기법[5]이 있으나 이 기법 역시 시스템 셋업시에 한번만 설정하는 단점이 있기 때문에 셋업 후 나타나는 시스템 변화를 반영할 수 없게 된다. 이 연구에서는 실시간 일정계획 기법과 혼합 일정계획 기법의 장점을 이용하여, 일정계획 구간을 결정하는 부분에서는 실시간 일정계획 기법을, 할당규칙을 적용하는 부분에서는 혼합 일정계획 기법을 사용하여 실시간 혼합 일정계획 기법을 제안한다.

본 장에서는 실시간 일정계획 알고리즘과 혼합 일정계획 알고리즘에 대하여 알아보도록 한다.

### 2.1 실시간 일정계획 알고리즘

<그림 1>은 실시간 일정계획 알고리즘과 다른 유연생산시스템 컨트롤러간의 관계들을 보여주고 있다. 시장의 변화와

기업의 경영정책, 그리고 그 밖의 여러가지 외부요인들이 공장제어시스템(Plant control system)의 의사결정 지원시스템(Decision Support System)으로 들어와서 다음과 같은 정보를 설정한다.

- 부품 방출 정보
- 부품 정보(납기일, 부품 가공순서, 가공시간 등)
- 수행척도
- 후보할당규칙

위의 정보를 기초로 하여 실시간 일정계획 알고리즘은 다음 일정계획 구간동안에 최적의 수행척도를 가지는 할당규칙을 선택한다. 선택된 할당규칙을 감독자 통제 시스템(Supervisory control system)이 받아서 공장제어시스템에서 가져온 부품정보와 함께 유연생산시스템 설비를 제어하게 된다. 이 실시간 일정계획 알고리즘과 감독자통제시스템, 그리고 각각의 유연생산시스템의 설비들은 항상 온라인상태로 예상하지 못한 사건이나 현재 각 설비의 상태에 대한 정보를 서로 공유하게 된다.

그러나 실시간 일정계획 알고리즘들이 시스템 전체에서 단일규칙을 사용하는 것 이상으로 시스템의 수행도를 높일 수 있는 잠재력을 가질 수는 있으나, 그러한 알고리즘에는 다음과 같은 어려움이 존재한다.

- (1) 현재의 할당규칙을 적용하는 일정계획 구간에서 다음 할당규칙을 적용하는 일정계획 구간을 정의하기는 힘들다.

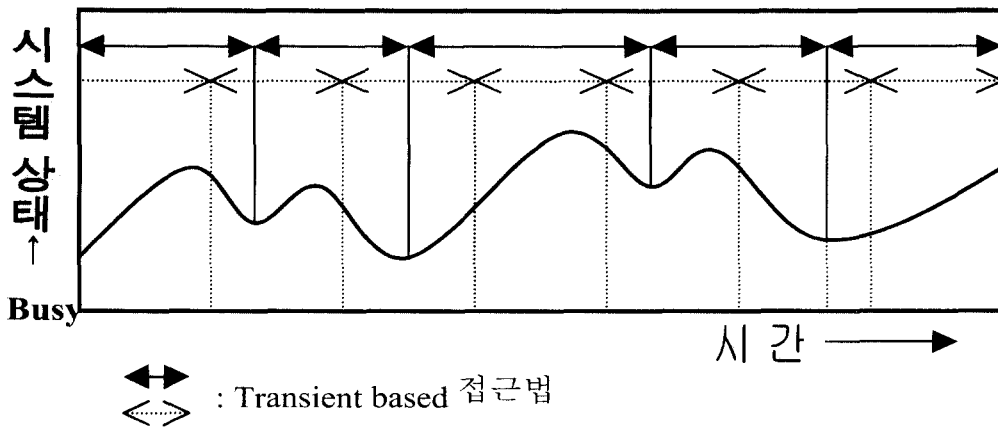


그림 2. Transient-based 접근법과 Constant interval 접근법에 의한 일정계획 구간설정.

(2) 다음 일정계획 구간에 대하여 후보 할당규칙의 수행도를 계산하기가 어렵다.

이러한 어려움을 해결하기 위하여 이 연구에서 제시하는 알고리즘은 시스템 가동시간에 대한 시간창(time window: 시간영역)을 이용하는 것인데, 시간의 흐름에 따라서 시스템 일정계획 구간을 설정하고 그 상태를 점검하는 방식이다. 시간창을 이용한 방법에는 constant interval 접근법과 transient-based 접근법이 있다.

Constant interval 접근법이란 전체 시스템 가동시간을 관리자가 임의의 시점들로 나누어서 일정시점에서 시스템의 상태를 점검하여 그것에 맞는 할당규칙을 적용하는 방법론이다. Transient-based 접근법은 이 연구에서 이용하는 방법론으로서 불편(unbiased) 할당규칙을 모의실험을 통하여 적용한 후 나온 결과를 가지고 시스템 가동시간을 단기간(short term period)으로 나눈 후 각각의 단기간의 시작점에서 새로운 할당규칙을 적용하는 방법론이다. 여기서 불편 할당규칙이란 어떤 시스템 환경에서도 수행척도면에서 거의 유사한 값을 가지는 할당규칙을 말한다. Constant interval 접근법과 transient-based 접근법에 대한 예를 <그림 2>에서 보여준다. 그러나 Constant interval 접근법은 시스템 상태의 변화를 모두 수용하기 힘들기 때문에 유연생산시스템과 같이 실시간으로 설비와 환경이 변화하는 시스템 내에서는 적용시키는 데 문제점이 많이 발생한다. 즉, 유연생산시스템과 같은 동적시스템에서 실시간 일정계획 알고리즘에 대한 일정계획 구간의 길이는 시스템 상태를 기준으로 하여 동적으로 변화할 수 있어야만 한다. 그렇기 때문에 이 연구에서 제시하는 알고리즘은 transient-based 접근법을 기초로 하였다. 그러나 Transient-based 접근법에서도 한 가지 중요한 문제점이 있다. 가령 어떤 부품이 연이은 두 개의 일정계획 구간에 걸쳐서 작업을 받는다고 하자. 그렇다면 이 부품은 이 두개의 일정계획 구간을 설정하고, 할당규칙을 선택하는 데 있어서 잘못된 오류를 유발시킨다. 즉, 일정계획 구간이 끝났는데도 불구하고 작업장에 남아있는 부품은 할당규칙을 선택하는데 있어서 부적합한 결과를 유발시키게 된다. 바로 이런 문제를 censored 데이터 문제라고 부른다(Kiran and Smith, 1984).

이 문제에 대한 완벽한 해결방안을 위하여 제시된 연구는 아직 없다(Kiran and Smith, 1984). 그러나 본 연구에서는 3장에서 이 문제를 고려한 알고리즘을 제시한다.

## 2.2 혼합 할당규칙

시스템 내의 각각의 작업장마다 여러 가지 다른 할당규칙을 적용시키는 일정계획 기법을 혼합 할당규칙(Mixed Dispatching Rule)이라고 한다. 다시 한번 정의하자면 모든 기계에 하나의 할당규칙을 사용하는 방법이 아닌 각각의 기계에 각각의 할당규칙을 배분하는 알고리즘을 말하는 것이다. 이 알고리즘의 배경은 유연생산시스템에서 종류가 다른 각각의 부품들은 각기 다른 가공순서와 가공시간을 가지기 때문에 각각의 대기열에서 한 기계의 상황과 다른 기계의 상황이 다르며 서로 영향을 준다는 점에서 비롯되었다.

일반적으로, 하나의 할당규칙을 사용하는 방식에서는 비교하고자 하는 할당규칙의 수만큼 모의실험함으로써 가장 성능이 뛰어난 규칙을 평가할 수 있다. 그러나 각각의 기계에 다른 할당규칙을 적용하는 방식에서는 후보가 되는 할당규칙들을 각각의 기계에 할당함으로써 만들어지는 할당규칙의 조합 중 성능이 뛰어난 조합을 골라낼 필요가 있다. 그러나 이 조합을 골라내는데 있어서 많은 모의실험의 계산이 필요하게 되는 데, 가령  $m$ 개의 기계에  $n$ 개의 할당규칙을 적용하기 위한 총 모의실험 횟수는  $m \times n$ 번이 된다. 이 경우 실제로 실시간 일정계획 시스템에서 혼합 할당규칙을 적용하는 것은 비실용적인 방법이 된다. 그래서 본 연구에서는 이 조합을 골라내는 데 필요한 시간을 줄이기 위한 방법을 제 3장에서 제시한다.

## 3. 실시간 혼합 일정계획 알고리즘

이 장에서는 실시간 혼합 일정계획 기법에 대하여 설명하고자 한다. 이 실시간 혼합 일정계획 알고리즘은 일정계획 구성요소(실시간 일정계획 부분)와 할당규칙 선택(혼합 일정계획 부분), 모의실험 부분 등 3가지 단계로 결정된다. 이 알고리즘을 분석

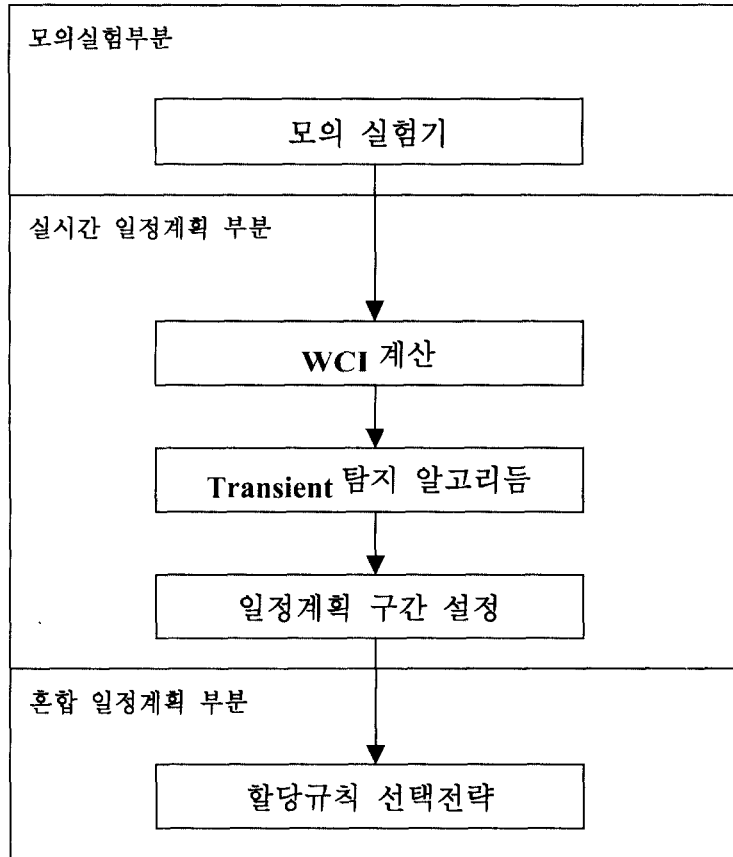


그림 3. 실시간 혼합 일정계획 개략도.

하여 보면, 먼저 모의실험은 시스템 상태를 평가하는 Work-Center Index(WCI)를 계산하기 위하여 필요한 정보를 보내준다. 그 다음은 이 WCI를 최소 일정계획 구간(Minimum Scheduling Interval:MSI) 단위로 그룹화시킨 후 transient 탐지를 하여 일정 계획 구간을 결정한다. 마지막으로 일정계획 구간이 설정되면 현재 작업장 상태를 향상시킬 수 있는 최적의 할당규칙 조합이 선택알고리즘에 따라서 모의실험을 통하여 선택된다. 결국 앞에서 결정한 일정계획 구간에서 이 할당규칙 조합이 계속 적용된다.

<그림 3>은 이 내용을 도식화시켜 놓은 것이다.

3.1절에서는 알고리즘에 사용되는 기호를 정의하였고, 3.2절에서는 알고리즘을 제시하였다.

3.1 기호정의

- $DT_j$ : 부품  $j$ 의 납기일
- $RDT_j$ : 남아 있는 납기일  
(부품  $j$ 의 납기일 - 부품  $j$ 가 시스템에서 소요한 총 시간)
- $N$ : 시스템 안에 있는 총 부품수
- $n$ : 시간  $t$ 에서 한 작업장으로 들어오는 모든 부품 수, 즉 그 작업장의 대기열 안에 있는 부품수 + 그 작업장으로 향하는 AGV 안에 있는 부품수 + 그 작업장으로 이동하기 위하여AGV 대기열 안에 있는 부품수
- $PT_j$ : 부품  $j$ 가 어떤 작업장에서 가공되는 가공시간

- $TR_j$ : 부품  $j$ 가 어떤 작업장으로 들어가기 위하여 이동한 시간  
AGV 대기시간 포함(모의실험으로 예측)
- $WT_j$ : 부품  $j$ 가 가공되기 위하여 기다리는 시간(모의실험으로 예측)
- Work Center Index (WCI): 어떤 작업장의 상태를 점검하기 위한 수치
- $t$ : WCI가 계산되는 시점에서의 모의실험 시간

3.2 일정계획 구간 설정

다음 일정계획 구간은 현재시점에서 <그림 2>에 나타나 있는 것처럼 다음 '상향' transient 상태의 시작점까지이다. 이런 종류의 정의는 구간 끝점에 남아 있는  $n$ 의 수가 구간 내의 최고값보다 훨씬 작기 때문에 어느 정도 censored 데이터 문제를 해결할 수 있다(Ishii and Talavage, 1991). WCI는 혼잡도, lateness, 흐름 시간 등을 한꺼번에 보여주는데, transient상태도 이 WCI의 시간열 데이터의 분석을 통하여 형성된다.

3.2.1 작업장상태에 대한 척도

본 연구는 수행척도의 평균값(means)을 통하여 transient상태를 결정한다. 기존연구들은 앞서 언급한 혼잡도, lateness, 흐름 시간 등을 수행척도로 사용하였으나 이것들은 모두 시스템 전체를 고려하는 경우이기 때문에 본 연구목적에 적당하지 않다. 그래서 이 연구에서는 각 작업장의 상태를 측정하기 위하여

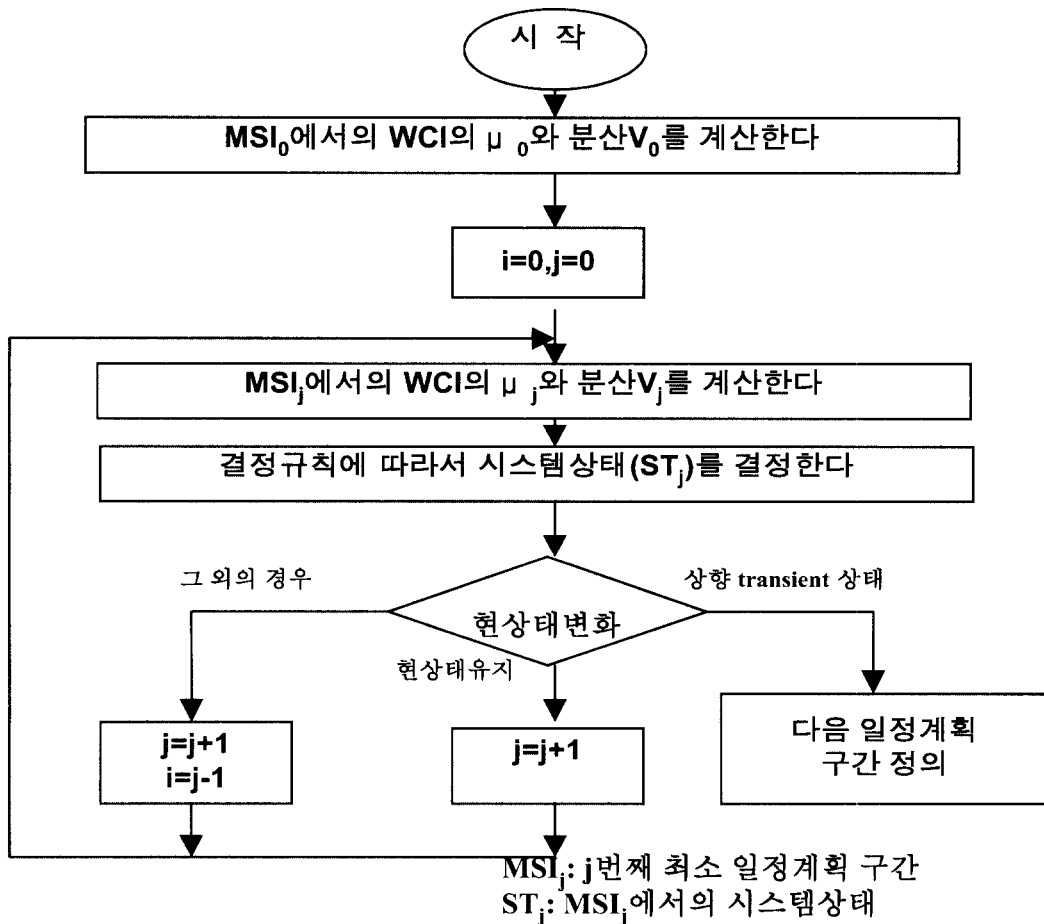


그림 4. Transient 탐색 알고리즘.

WorkCenter Index(이후 WCI라고한다)라고하는 기준값을 사용하였다. WCI는 식 1과 같이 정의된다.

$$WCI(t) = \sum_{j=1}^n \frac{PT_j + WT_j + TR_j}{n \times RDT_j} \quad (1)$$

이 WCI를 분석하여 보자. 한 작업장에 들어오는 각 부품들이 그 작업장에서 소요하는 시간을 그 부품수로 나눈 수치는 곧 그 작업장에서 작업을 하기 위하여 소요하는 평균 체류시간이 된다. 이것을 다시 각 부품마다 남아있는 납기일의 함으로 나누게 되면 납기에 민감한 작업장의 변동을 나타내게 된다. 이 WCI의 주목할만한 특징은 2가지가 있다. 첫 번째가 평균 체류시간이 길어지면 그것은 곧 WCI를 증가시킨다. 왜냐하면 작업장의 용량이 초과되고 각 부품들의 대기시간이 증가되어 WCI도 증가하기 때문이다. 두 번째는 납기에 촉박해진 부품이 작업장안으로 들어오면 WCI의 분모에 있는 RDT<sub>j</sub>값을 감소시키기 때문에 WCI를 증가시킨다.

WCI를 계산하기 위하여 모의실험기(simulator)는 일정계획 구간 동안의 각 작업장으로 들어오는 부품들에 대하여 대기시간과 이동시간을 예측하여야 한다. Transient 구간의 끝점을 예

측하기 위하여 모의실험기는 처음에는 각 작업장 모두에게 같은 할당규칙을 적용시켜야 한다. 본 연구에서는 선입선출(FIFO)규칙을 이 목적을 위하여 사용하였다. WCI의 주된 목적은 작업장의 변동을 예측하기 위한 것이기 때문에 선입선출규칙은 이 목적을 위하여 상대적으로 가장 불편(unbiased)한 할당규칙이다(Blackstone, et al., 1982). WCI에 대한 시간열 데이터는 다음과 같이 정의되는 구간 각각에서 계산한다.

- WCI를 계산하는 구간 = 최소 일정계획 구간(minimum scheduling interval)/샘플링숫자
- 샘플링 숫자 = 10에서 15 사이의 정수

최소 일정계획 구간(minimum scheduling interval)은 모든 부품 중에서 가장 짧은 총 가공시간과 같다. 이 최소 일정계획 구간은 다음절에서 더욱 자세히 다룬다.

### 3.2.2 Transient 탐지 알고리즘

Transient 상태는 WCI의 시간열 데이터의 분석에 의해서 정의된다. <그림 4>는 기존의 Ishii and Talavage(1996)에서 제시

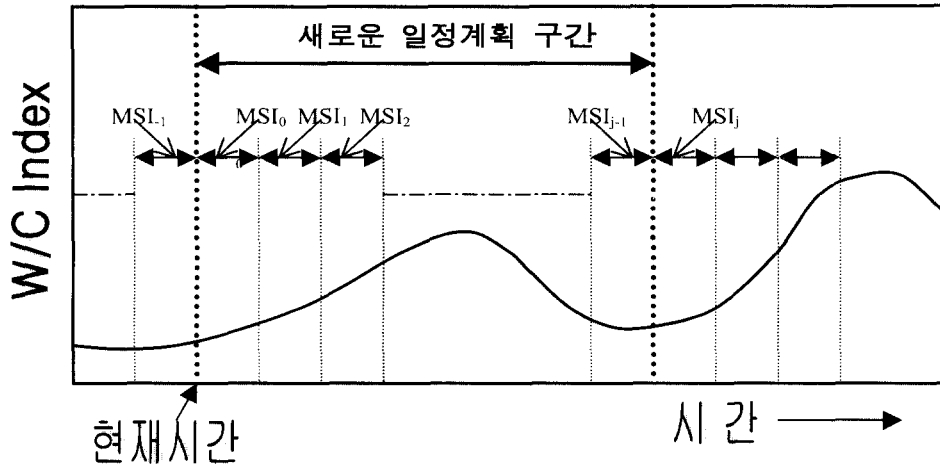


그림 5. 최소 일정계획 구간(MSI).

한 transient 구성 알고리즘을 보여주며, 본 연구에서도 이 알고리즘을 이용하였다. 이 알고리즘에서 WCI의 시간열 데이터는 <그림 5>에서 보여주는 것과 같이 최소 일정계획 구간(minimum scheduling interval: MSI)으로 나뉜다.

이 MSI는 WCI의 평균과 분산을 계산하기 위하여 사용된다. 작업장의 변화들은 MSI를 계속해서 비교하므로 탐지될 수 있다. <그림 4>에서 MSI<sub>j</sub>, μ<sub>j</sub>, V<sub>j</sub>는 각각 j번째 최소 일정계획 구간, MSI<sub>j</sub>안에서의 WCI의 평균, MSI<sub>j</sub>안에서의 WCI의 분산을 나타낸다.

이 알고리즘에서 MSI<sub>j</sub>의 작업장 상태는 다음과 같은 결정규칙을 기준으로 정의된다. 이런 규칙들에서 ST<sub>i</sub>는 i번째 구간에서 작업장의 상태를 가르쳐 준다.

시스템 상태를 결정하는 결정규칙:

- (1) 만일 ST<sub>i</sub>가 안정상태이고 μ<sub>i</sub> = μ<sub>j</sub>이면 ST<sub>i</sub>는 안정상태이다.
- (2) 만일 ST<sub>i</sub>가 안정상태이고 μ<sub>i</sub> > μ<sub>j</sub>이면 ST<sub>i</sub>는 하향 transient 상태이다.
- (3) 만일 ST<sub>i</sub>가 안정상태이고 μ<sub>i</sub> < μ<sub>j</sub>이면 ST<sub>i</sub>는 상향 transient 상태이다.
- (4) 만일 ST<sub>i</sub>가 상향 transient 상태이고 μ<sub>i</sub> < μ<sub>j</sub>이면 ST<sub>i</sub>는 상향 transient 상태이다.
- (5) 만일 ST<sub>i</sub>가 상향 transient 상태이고 μ<sub>i</sub> = μ<sub>j</sub>이면 ST<sub>i</sub>는 안정상태이다.
- (6) 만일 ST<sub>i</sub>가 상향 transient 상태이고 μ<sub>i</sub> > μ<sub>j</sub>이면 ST<sub>i</sub>는 하향 transient 상태이다.
- (7) 만일 ST<sub>i</sub>가 하향 transient 상태이고 μ<sub>i</sub> > μ<sub>j</sub>이면 ST<sub>i</sub>는 하향 transient 상태이다.
- (8) 만일 ST<sub>i</sub>가 하향 transient 상태이고 μ<sub>i</sub> = μ<sub>j</sub>이면 ST<sub>i</sub>는 안정상태이다.
- (9) 만일 ST<sub>i</sub>가 하향 transient 상태이고 μ<sub>i</sub> < μ<sub>j</sub>이면 ST<sub>i</sub>는 상향 transient 상태이다.

ST<sub>i</sub>의 안정상태를 확인하기 위하여 이 연구에서는 기존연구와는 다르게 통계적 기법인 Behrens and Fisher(1990) 방법을 이

용하였다. 각 MSI간의 분산의 차이를 알 수 없기 때문에 귀무가설 μ<sub>i</sub> = μ<sub>j</sub> 를 검정할 수 없다. 그렇기 때문에 Behrens-Fisher는 식(2)을 통하여 통계량 t\*<sub>0</sub>를 t분포에 근사시켜 귀무가설 μ<sub>i</sub> = μ<sub>j</sub> 을 검정하였다.

$$t^*_{0} = \frac{\bar{X}_i - \bar{X}_j}{\sqrt{\frac{S_i^2}{n_i} + \frac{S_j^2}{n_j}}} \quad (2)$$

t\*<sub>0</sub>는 식 (2)를 이용하여 구한 자유도를 가지고 t분포에 근사적으로 분산되어 있다.

이 값은 귀무가설을 테스트하는 통계량이다. 만일 t\*<sub>0</sub>가 0.18 이고 자유도가 16이라면 유의 수준 5%에서 t\*<sub>0</sub> - t<sub>0.5, 16</sub> 이며 이것은 귀무가설 i = j 을 기각할 수 없게 된다.

$$v = \frac{\left(\frac{S_i^2}{n_i} + \frac{S_j^2}{n_j}\right)^2}{\frac{(S_i^2/n_i)^2}{n_i+1} + \frac{(S_j^2/n_j)^2}{n_j+1}} - 2 \quad (3)$$

만일, 현 시스템 상태가 안정상태나 하향 transient 상태에서 상향 transient 상태로 변화한다면, 일정계획 구간은 현시점에서부터 모의실험을 통하여 다음 상향 transient 상태가 탐지될 때까지이다. <그림 5>에서 나타난 것처럼 MSI<sub>i</sub>의 시작점에서 상향상태가 시작되므로, 다음 일정계획 구간은 현시점(MSI<sub>0</sub>)에서 MSI<sub>i</sub>의 시작점까지로 정의할 수 있다. 만일 MSI<sub>j</sub>에서 시스템 상태가 상향 transient 상태로 변화하지 않는다면 다음 MSI의 시스템 상태를 점검한다. 다음 일정계획 구간이 정의되면 이 알고리즘은 종료되며 모의실험과 WCI계산도 종료된다.

### 3.2.3 혼합 일정계획 구간 설정

앞에서 설명한 일정계획 구간은 각각의 작업장의 상태에 따라서 개별적으로 정의하는 것이다. 이제는 일정계획 구간을 보는 관점을 시스템 전체로 확대하여 어떤 시점에서 각 작업

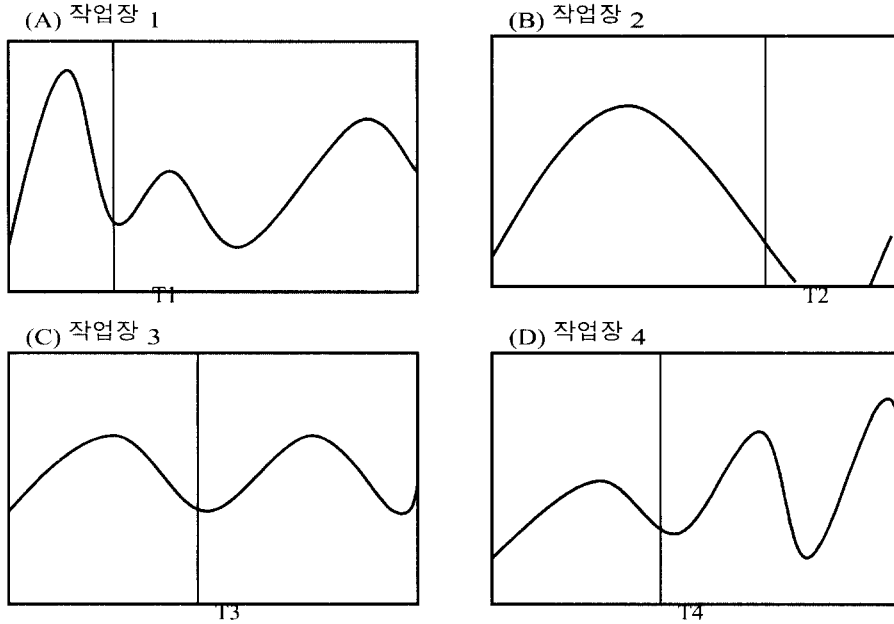


그림 6. 혼합 일정계획 구간 설정의 예.

장의 할당규칙을 전환시켜야 하는지를 정의하여야 한다. <그림 6>의 예를 보도록 하자.

<그림 6>의 A, B, C, D는 시간에 따른 WCI를 계산한 수치별 그래프화(y축: WCI)시킨 것이다. 여기서 각 작업장의 일정계획 구간은 T1, T2, T3, T4가 가리키는 것처럼 다르다. 여기서 우리가 처음으로 일정계획 구간을 결정하는 것은 두 가지 방법이 있다.

방법 1 : 각 작업장 그래프의 첫 번째 일정계획 구간 T1, T2, T3, T4에 대한 평균값을 구하여 그 시점까지를 시스템의 첫 번째 일정계획 구간으로 정한다

방법 2 : 각 작업장 그래프의 첫 번째 일정계획 구간 T1, T2, T3, T4중에서 최소값을 가지는 시점을 구하여 그 시점까지를 시스템의 첫 번째 일정계획 구간으로 정한다.

제 4장에서 설명할 모의실험 과정에서는 방법 1, 2를 각각 수행하여 그 결과값을 알아 보도록 한다.

### 3.3 할당규칙 선택요소

이 요소에서는 transient 탐색과 일정계획 구간 설정에 의하여 결정된 새로운 일정계획 구간에서 사용할 할당규칙을 선택한다. 이 요소에서 사용할 후보 할당규칙들은 단일 할당규칙을 사용하며, 각 후보 할당규칙은 모의실험기에 의하여 제공되는 모의실험 결과로 계산되어진다. 할당규칙의 선택 기준은 새로운 일정계획 구간에서 수행 척도(Performance criteria)를 최고로 만족하는 것이다. 그러나 각 후보의 성과를 계산하기에는 일정계획 구간이 짧기 때문에 이 작업을 위하여 몇 가지 전략이

필요하다. 동시에 앞에 2장에서 언급하였던 censored 데이터 문제도 여기서 최소화시킨다.

이 연구에서는 할당규칙을 선택하는 데 있어서 2가지의 전략을 제시한다. 두 가지 모두 censored 데이터 문제를 해결하기 위하여 Ishii and Talavage가 제시한 불편(unbiased) 할당규칙을 문제가 되는 구간에 투입한다.

#### 3.3.1 할당규칙 선택 전략 1(그림 7 참조)

Step 1 : 혼합 일정계획 구간은 방법 1을 사용하여 구하며, 모의실험구간(T1)은 현재 시점에서부터 일정계획 구간 안에 들어온 부품이 그 작업장 밖으로 빠져나갈 때 까지로 정의한다.

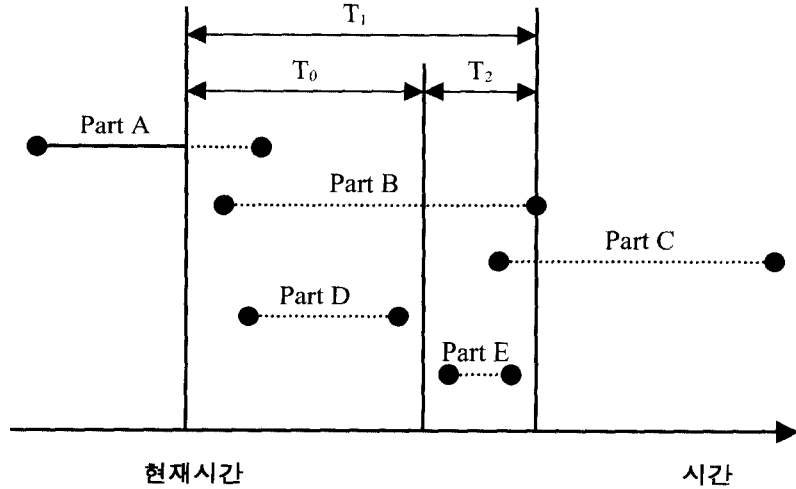
Step 2 : 모든 작업장에 선입선출(FIFO) 할당규칙을 적용하여 거기에 따른 결과로 일정계획 구간(T0)을 설정한다.

Step 3 : 일정계획 구간의 끝점부터 아직 작업장에 남아 있는 부품들이 작업을 완료할 때까지 구간을 설정(T2)한다.

Step 4 : 후보 할당규칙을 모두 각 작업장에 적용시키고 일정계획 구간의 끝점 이후에도 각 작업장에 남아 있는 부품들에 대하여는 선입선출(FIFO: 선입선출은 새 일정계획 구간 이후의 부품이동에 대하여 불편(unbiased) 할당규칙이기 때문이다) 할당규칙을 적용한 후에 전체 시스템의 평균 흐름시간과 지연시간을 구한다.

Step 5 : 각 작업장에 대하여 전체 시스템의 평균 흐름시간과 지연시간을 이용하여 할당규칙순위를 정하고, WCI 값이 가장 높은 작업장에 가장 높은 순위를 가지는 할당규칙을 적용한 후 다시 시스템의 평균 흐름시간과 지연시간을 계산한다.





$T_0$  : 다음 일정계획 구간  
 $T_1$  : 총 모의실험 구간  
 $T_2$  : censored 데이터를 보정하기 위한 추가 구간

그림 7. 전략1에 대한 모의실험 구간.

Step 6 : Step 5를 반복수행하여 시스템의 평균 흐름시간과 지연시간에 대한 최적의 할당규칙의 조합을 선택한다.  
 Step 7 : 새 일정계획 구간의 할당규칙 조합이 결정되면 다음 일정계획 구간에 Step 1 적용

Step 4 : 새 일정계획 구간의 할당규칙 조합이 결정되면 다음 일정계획 구간에 Step 1 적용  
 이 알고리즘은 언제나 Step 1에서 모의실험 구간을 현 시점에서부터 시스템종료 시점까지로 설정하기 때문에 여기서 구한 할당규칙 조합은 단일 할당규칙을 적용할 때 보다 더욱 좋은 수행도가 나온다는 것을 보장한다.

<그림 7>은 Step 2에 대하여 좀 더 명확한 이해를 도울 수 있다.  $T_0$ 는 작업장상태를 기준으로 하여 정의된 다음 일정계획 구간이다. 구간  $T_0$ 에 존재하는 부품은 A와 B, D이다. 그러나 부품 B의 작업이 완료되지 않은 시점에서  $T_0$ 구간이 끝나기 때문에 작업장에는 아직 부품 B가 남아 있다. 이 경우 후보할당규칙들의 수행도는 부품 B의 작업 완료시점( $T_1$ )까지 계산해야 하며, 결국 여기에 대하여 censored 데이터 문제가 발생한다. 이 censored 데이터 문제를 해결하기 위하여 구간  $T_0$ 에서는 후보할당규칙을 적용하고, 구간  $T_2$ 에서는 FIFO를 적용하여 수행도를 계산한다. 그리고 현 일정계획 구간이 끝난 다음 일정계획 구간부터는 부품 C, E 등을 가지고 새 일정계획 구간을 정의하는 것이다. <그림 8>은 전략1을 개략적 순서도로 나타내었다.

4. 실험 및 결과분석

이 장에서는 연구와 단일 일정계획 알고리즘, 그리고 실시간 일정계획 알고리즘에 대한 수행도를 유연생산시스템 모델을 통하여 실험하였다. 거의 대부분의 실험환경과 데이터는 기존 연구에서 이용한 것과 거의 일치시켰다.

4.1 유연생산시스템 모델

먼저 모의실험을 위하여 개발툴로서는 <그림 9>에 나와 있는 것처럼 Arena 3.0을 이용하였으며, 이 실험을 보다 정확히 반영하기 위한 Application Interface로는 Visual Basic 5.0을 이용하였다.

3.3.2 할당규칙 선택전략 2  
 Step 1 : 혼합 일정계획 구간은 방법 2로 설정하며, 할당규칙을 계산하기 위한 모의실험 구간( $T_1$ )은 현 시점에서부터 시스템 종료시점까지로 설정한다.  
 Step 2 : 각 작업장에 대하여 전체 시스템의 평균 흐름시간과 지연시간을 이용하여 할당규칙순위를 정하고, WCI 값이 가장 높은 작업장에 가장 높은 순위를 가지는 할당규칙을 적용한 후 다시 시스템의 평균 흐름시간과 지연시간을 계산한다.  
 Step 3 : Step 2를 반복수행하여 시스템의 평균 흐름시간과 지연시간에 대한 최적의 할당규칙의 조합을 선택한다.

4.1.1 작업환경  
 <그림 9>는 본 연구의 실험을 위한 유연생산시스템의 물리적 배치를 보여준다. 이 시스템은 2개의 L/U(Load/Unload) 작업장과 4개의 작업장, 3개의 AGV로 구성되어 있다.  
 각각의 L/U와 작업장들은 캐루셀(carousel) 보관시스템을 가지고 있다. 모든 부품은 L/U1으로 시스템에 들어와서 모든 작업이 끝나면 L/U2로 나간다. 부품들이 작업장들 사이에서 이동

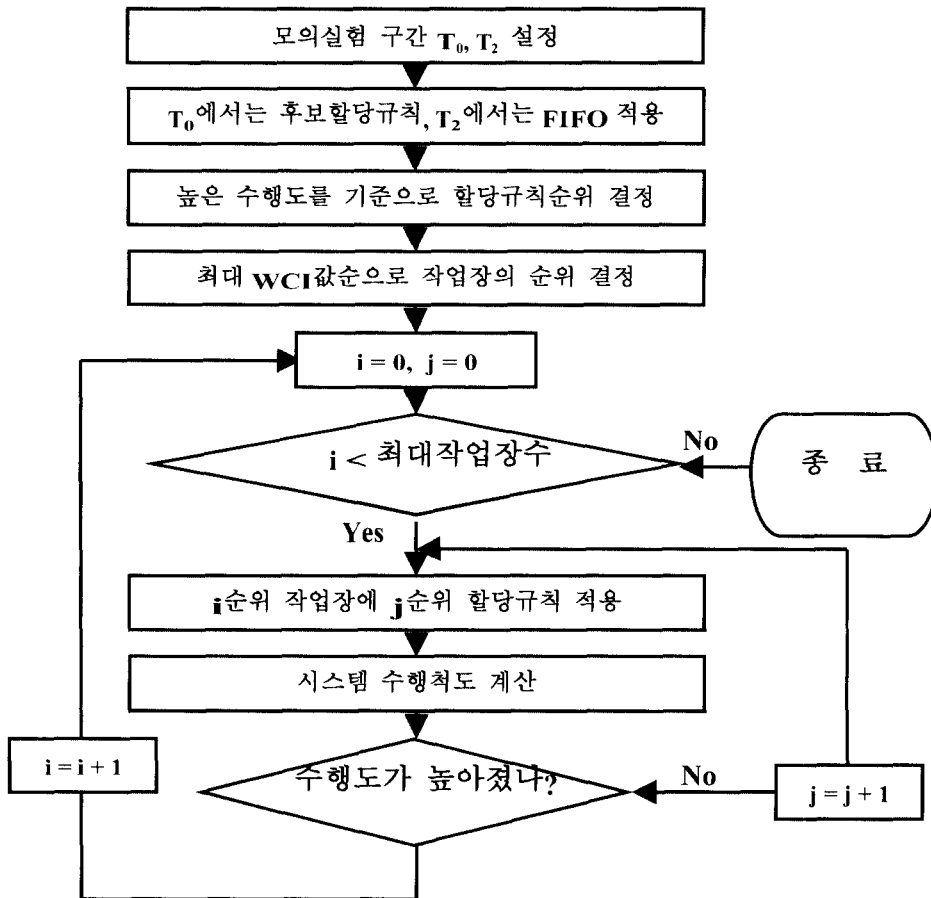


그림 8. 혼합 할당규칙에 대한 선택 알고리즘.

할 때는 AGV를 통해야만 한다.

자재처리(Material Handling)에 대하여는 3대의 AGV가 작업장들간의 운송을 담당한다. AGV의 보다 상세한 정보는 다음과 같다.

- AGV 속도: 15.0 m/min(부품을 실었을 때)  
16.5 m/min(부품을 실지 않았을 때)

- AGV 길이: 1.8 m
- AGV 최대용량: 부품 1개 / AGV
- AGV는 작업장 안에서 일반통행(단일방향이동)만 할 수 있다.

그리고 각 작업장과 L/U들의 대기열 길이는 다음과 같다.

- 가공 전: 6부품
- 가공 후: 4부품

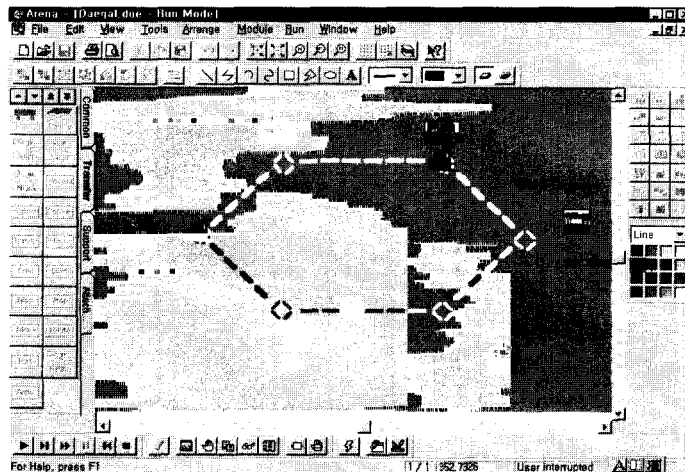


그림 9. 실험의 물리적 배치.

4.1.2 실험에 필요한 데이터

이 모델에는 6개의 부품이 있다. 가공순서는 부품마다 다르며 각 작업장에 대한 가공시간도 모두 다르다. 추가로 각 부품은 저마다의 납기일정이 정해져 있으며 이 모든 사항에 대하여는 <표 1> 과 <표 2>에 자세히 나와 있다.

표 1. 부품 가공순서

순서 부품	1	2	3	4	5	6
A	L/U 1	W/C 1	W/C 4	W/C 3	L/U 1	***
B	L/U 1	W/C 2	W/C 1	W/C 3	W/C 1	L/U 1
C	L/U 1	W/C 2	W/C 1	W/C 3	W/C 1	L/U 1
D	L/U 1	W/C 3	W/C 2	W/C 1	L/U 1	***
E	L/U 1	W/C 4	W/C 2	W/C 1	L/U 1	***
F	L/U 1	W/C 4	W/C 3	W/C 2	W/C 1	L/U 1

W/C: 작업장(Work Center)  
\*\*\* 작업이 없음

표 2. 부품 가공시간 및 납기

작업 부품	단위: 분						납기
	W/C 1	W/C 2	W/C 3	W/C 4	L/U 1	L/U 2	
A	10.00	***	18.00	20.00	9.00	5.00	180.00
B	23.00	15.00	15.00	21.00	8.00	5.00	220.00
C	23.00	15.00	15.00	32.00	8.00	5.00	180.00
D	10.00	20.00	22.00	***	7.00	5.00	150.00
E	15.00	16.00	***	18.00	12.00	5.00	200.00
F	28.00	21.00	17.00	19.00	15.00	5.00	310.00

\*\*\*작업이 없음

예를 들어 부품 A의 가공순서는 L/U 1으로 시스템 내에 들어와서 작업장 1, 4, 3을 거친 후 L/U 2를 통하여 시스템 밖으로 나가게 된다.

모의실험 시간 및 구간에 대하여는 전체 제조시간(총 일정계획 시간) : 2880분(2일)으로 하였으며, Transient based 실시간 일정계획을 위한 최소 일정계획 구간(MSI)은 62분(부품 A)으로 한다.

4.1.3 기타 실험정보

실험에서 이용한 후보 할당규칙들은 다음과 같다.

- NINQ: 다음에 가야하는 작업장의 대기열이 가장 짧은 부품을 선택하여 우선순위를 가장 높게 준다.
- SPT: 가장 짧은 작업시간을 가진 부품을 선택하여 우선순위를 가장 높게 준다.
- EDD: 가장 빠른 납기를 가진 부품을 선택하여 우선순위

를 가장 높게 준다.

- FIFO: 작업장의 대기열에 가장 먼저 들어온 부품을 선택하여 우선순위를 가장 높게 준다.

다음은 이 에서 이용한 수행적도를 계산하는 식이다.

(1) 평균 흐름 시간

$$\sum(\text{부품 } i \text{의 완성시간} - \text{부품 } i \text{의 시스템진입시간}) / n$$

(2) 평균 납기지연 시간

$$\sum \max(0, \text{부품 } i \text{의 지연시간}) / \text{지연부품수}$$

$$\text{부품 } i \text{의 지연시간} = \text{부품 } i \text{의 완성시간} - \text{부품 } i \text{의 시스템진입시간} - \text{부품 } i \text{의 납기}$$

부품투입은 기존연구에서 실시한 것처럼 다음과 같은 3가지 방식으로 설정한다.

- 실험 A: 지수 분포(평균 10분)
- 실험 B: 지수 분포(평균 15분)
- 실험 C: 무작위 방출(Random Release)

전략 1과 2를 모두 포함하여 예비실험을 20회 시행하여 수행도를 측정하였다. 그 중에서 대표적인 수행도인 평균 흐름시간을 고려하여, 통계적인 실험반복 횟수를 결정하였다.

예비실험으로 평균 흐름시간의 평균값은 154.97(전략 1, 2포함)이고, 표준편차는 16.08인 결과를 얻었으며, 본 연구에서는  $\alpha$ 를 0.05로 상대정도(relative precision)  $\gamma$ 를 0.05로 상정하여 실험횟수를 산정하였다.

일반적인 실험횟수  $i$ 를 결정하는 과정은 식 (4)와 같다.

$$n^*(\gamma) = \min \left\{ i \geq n : \frac{t_{i-1, \alpha/2} \sqrt{s^2(n)/i}}{|X(n)|} \leq \gamma \right\} \quad (4)$$

$$\overline{X(n)} = 154.97$$

$$s^2(n) = 258.57$$

$$\gamma = 0.05$$

식 (4)를 만족시키는  $i$ 값은 20이다. 즉,

$$\frac{2.093 \times \sqrt{258.57} \div 20}{154.97} = 0.0486 < 0.05$$

이므로 실험횟수 20회를 사용하는 것이 통계적으로 유의하다고 판단되어, 각 실험의 최소 수행횟수를 20회로 결정하였다.

4.2 모의실험 수행결과 및 분석

실험 A, B, C를 각각 100회씩 모의실험한 결과가 <표 3>, <표 4>, <표 5>에 요약되어 있다. 각 표에서 괄호 안의 수치는 이 연구결과가 실시간 일정계획 알고리즘에 비하여 향상된 결과를 나타내고 있다.7

모든 실험방식에서 본 연구의 실시간 일정계획 알고리즘에 비하여 평균 흐름시간에서는 4.4%가 우수했으며 평균 납기 지연시간에서는 4.38% 향상하였다.

표 3. 실험방식 A의 결과

수행척도 할당규칙	평균 흐름시간	평균 지연시간
EDD	170.41	24.01
FIFO	178.38	23.86
NINQ	173.61	19.90
SPT	164.42	19.46
실시간 일정계획 이 연구 알고리즘	159.16(3.2%)	18.00(7.5%)
(전략 1)	(2.5%)	(0.09%)
이 연구 알고리즘 (전략 2)	154.54 (2.9%)	17.14 (4.8%)

표 4. 실험방식 B의 결과

수행척도 할당규칙	평균 흐름시간	평균 지연시간
EDD	167.87	22.86
FIFO	177.41	22.93
NINQ	162.21	18.15
SPT	167.90	19.13
실시간 일정계획 이 연구 알고리즘	158.56(2.25%)	17.55(3.3%)
(전략 1)	(2.9%)	(4.8%)
이 연구 알고리즘 (전략 2)	153.01 (3.5%)	16.24 (7.45%)

표 5. 실험방식 C의 결과

수행척도 할당규칙	평균 흐름시간	평균 지연시간
EDD	168.65	23.08
FIFO	179.90	26.49
NINQ	163.73	18.96
SPT	167.75	18.32
실시간 일정계획 이 연구 알고리즘 (전략 1)	159.46(2.61%) (3.27%)	14.03(2.34%) (4.1%)
이 연구 알고리즘 (전략 2)	153.72 (3.6%)	13.42 (4.34%)

이것은 단일 할당규칙에서 제일 좋은 결과를 낳은 SPT규칙에 대해 실시간 일정계획 알고리즘 향상률(3.11%, 2.68%)보다 더욱 좋아진 것이다. <그림 10>, <그림 11>은 각 수행척도에 대한 이 연구의 향상을 도식화한 것이다.

### 5. 결론

이 연구는 시시각각 변화하는 생산환경에 빠른 대처를 하기 위하여 실시간으로 일정계획을 수립하고 생산환경 중 작업장 변화를 정확하게 반영할 수 있는 알고리즘을 제시하였다. 이 알고리즘은 단일 할당규칙의 한계성을 극복하고, 시스템 내 부품의 흐름시간과 지연시간의 최소화를 위하여 작업장의 상태를 최대로 반영함과 동시에 납기에 민감할 수 있는 할당규칙조합을 선택하는 것이다.

이 연구의 특징은 첫 번째로 시스템 상황 중에서도 특히 작업장의 상태변화에 중점을 두어 다음 일정계획 구간에서 동적으로 할당규칙을 선택한다. 두 번째, 다음 일정계획 구간을 결

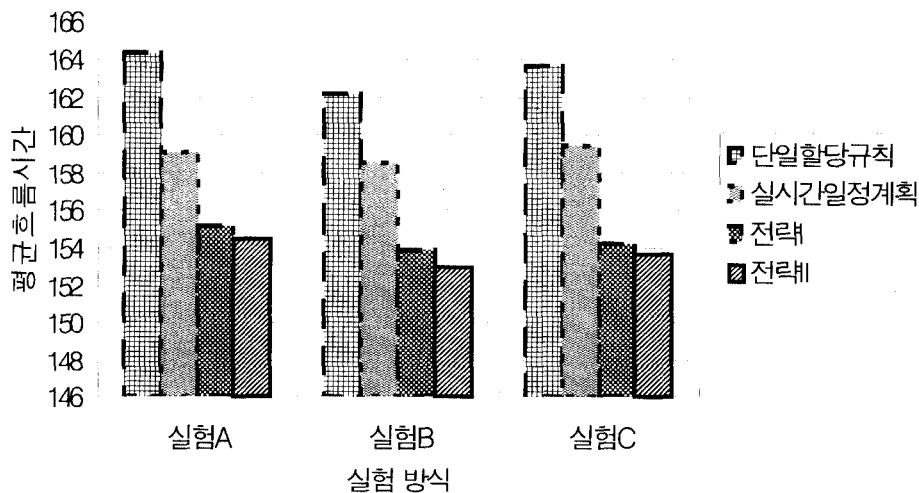


그림 10. 평균흐름시간의 향상 그래프.

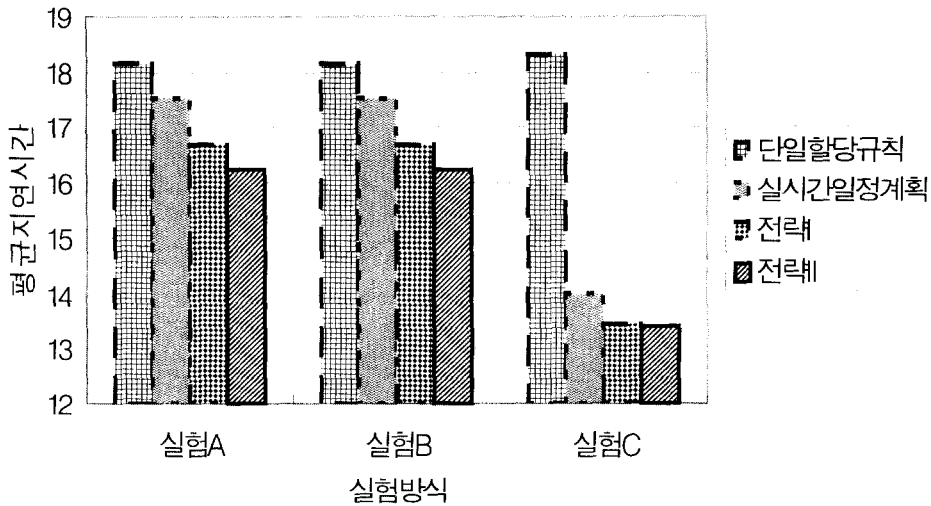


그림 11. 평균지연시간의 향상 그래프.

정하기 위하여 transient 상태를 점검하는 알고리즘을 포함하고 있으며, 세 번째는 최적의 할당규칙 조합을 선택하기 위한 2가지의 전략도 소개하고 있다.

모의실험은 각각의 전략과 실험방법에 맞추어 모두 같은 환경에서 실시하였으며, 이 연구에서 제시한 알고리즘의 성능은 4장에서 점검하였듯이 기존의 실시간 알고리즘보다 평균 흐름시간에서 4.4%, 평균 납기지연시간에서 4.38% 우수한 결과를 보여주었고, 단일 할당규칙을 적용하였을 때 보다 평균 흐름시간에 있어서 7.51% 뛰어난 결과를 보여주었다. 이것은 앞에서 언급한 것과 같이 각각의 작업장의 민감도를 알고리즘에 반영함으로써 작업장의 정체를 감소시키고 작업장간의 영향을 최대한으로 줄일 수 있도록 알고리즘을 설계하였기 때문이다.

추후 연구과제로는 첫째, censored 데이터 문제를 완전히 제거시킬 수 있는 명확한 방법론이 없기 때문에 여기에 따른 연구가 무엇보다 필요하다. 둘째, 혼합 할당규칙의 선택에 있어서 작업장수  $m$ 과 할당규칙의 수  $n$ 에 따른 계산량의 감소를 위한 알고리즘을 연구하는 것이 필요하다. 마지막으로 전문가 시스템을 통한 동적 할당규칙 알고리즘을 이 연구와 접목시켜서 알고리즘 전체 계산시간을 최소화시키는 연구가 필요하다.

참고문헌

노인규, 박성현(1996), 유연생산시스템에서 기계와 무인운반차의 할당

규칙에 관한 연구 한양대학교 대학원 산업공학과  
 Blackstone, J. H., Phillips, D. T. and Hogg, G. L. (1982), A state-of-art survey of dispatching rules for manufacturing job shop operations. International Journal of Production Research, 20, 27-45.  
 Huajie Liu and Jian Dong (1996), Develop A Neural Network-based Automated Dispatching Rule Selector, IERC Proceedings, 5th Industrial Engineering Research Conference.  
 Ishii, N. and Talavage, J. J. (1991), A transient-based real-time scheduling algorithm in FMS. International Journal of Production Research., 29(12), 2501-2520.  
 Ishii, N. and Talavage, J. J. (1996), A Mixed Dispatching Rule Approach in FMS Scheduling, International Journal of Production Research, 34(1), 69-87.  
 Kiran, A. S. and Smith, M. L. (1984), Simulation studies in job shop scheduling I & II, computers & Industrial Engineering, 8, 87-105.  
 Pierrelval, H. and Mebarki, N. (1997), Dynamic selection of dispatching rules for manufacturing system scheduling. International Journal of Production Research, 35(6), 1575-1591.  
 Ro. I. K. and Kim J. I. (1990), Multi-criteria operational control rules in Flexible Manufacturing Systems, International Journal of Production Research, 28(1), 47-63.  
 Thesen, A. and Lei, L. (1986), An 'expert' system for scheduling robots in a flexible electropolating system with dynamically changing workloads. Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing System: Operations Research Models and Applications. 555-566.  
 William, W. H. and Douglas, C. M. (1990), Probability and Statistics in Engineering and Management Science(Ames: John Wiley & Sons).  
 Wu, S. Y. D. and WYSK, R. A. (1989), An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing system. International Journal of Production Research, 27(9), 1603-1623.  
 Viswanadham, N. and Narahari, Y. (1992), Performance Modeling of Automated Manufacturing Systems(Ames: Prentice Hall).