

# 불확정제약조건의 추론에 의한 일반화된 job-shop의 일정계획<sup>※</sup>

정남기\*·정승영\*\*·서민수\*\*\*

## Reasoning of Uncertain Constraints in a Generalized Job-shop Scheduling<sup>\*</sup>

Namkee Chung\*·Seungyoung Chung\*\*·Minsu Suh\*\*\*

### 요 약

이 연구의 모형은, 자원(기계, 인력), 주문(작업)의 의미를 확장한 "일반화된 job-shop"이다. 이 모형의 제약조건을 파악한 후, 불확정 제약조건의 표현과 추론방법을 연구하였으며, 이를 위하여 일정계획 수립 기능들을 모듈화하였다. 불확정 제약조건에 따라 모듈들을 선택적으로 사용하기 위한 추론 과정은, 먼저, 납기만족이 우선이면 제약만족(CSP) 일정계획 모듈을 사용하고, 그렇지 않으면 디스패칭 모듈을 사용한다. 그리하여, 납기, 작업시간의 불확정성과 자원의 변동, 새 주문의 추가, 일정계획 목표의 변경등을 반영하는 일정계획 수립방법을 제시하였다. 이 논문은 제약만족 일정계획과 디스패칭 일정계획을 하나의 틀에서 함께 사용함으로써, 일정계획 목표의 변경이나 제조환경의 변화에 대처할 수 있음을 보인다.

Key words: Job-shop scheduling, CSP model, Dispatching

※이 연구는 한국과학재단 1996년 핵심전문연구비(과제번호:961-1012-084-1)의 지원으로 수행 되었습.

\*전남대학교 산업공학과

\*\*한국과학기술원 산업경영연구소 CIM 연구센터

\*\*\*Univ. of Toronto 산업공학과

## 1. 서 론

보통의 “표준 job-shop”의 일정계획문제에서는 공정의 작업우선순위 제약과 한 기계에서 동시작업 불가(不可) 제약만을 고려한다. 이와 달리, 이 연구에서는 “일반화된 job-shop”의 일정계획문제를 정의하여 보다 넓은 범위의 문제들을 포함시킨다.

“일반화된 job-shop”은 표준 job-shop은 물론, 프로젝트 일정관리, 설비 수리계획, batch 생산계획 등의 문제를 해결할 수 있도록, 자원(기계, 인력), 주문(작업)의 의미를 일반화 시킨 모형이다. job-shop의 일정계획과 프로젝트 일정관리는 본질적으로 큰 차이가 없음에도 별도의 해결 방법을 사용하는 경우가 많다. Job-shop에서 자원의 종류, 수량 등의 제약을 추가 시키고, 자원 활용의 평준화를 도모하면, 이것으로 프로젝트 일정관리가 가능해진다. 설비 수리계획은 작업자의 수, 작업가능일, 생산계획, 동시작업/연속작업/회피작업 등의 제약을 고려하여, 정해진 기한 내에 설비가 수리되도록 그 순서를 정하는 문제이다(기병성,1995).

일반화된 job-shop의 일정계획문제에는 생산환경의 조건뿐만 아니라, 경영관리를 위해 미리 정해진 기준들, 예를 들면, 납기만족, 교체횟수 감소, 공정재고 감소, 자원활용도 높임 등까지도 포함시킬 수 있다. 그리고 자원은 구체적으로, 기계, 작업자, 치공구, 저장장소, 운반구 등을 포괄하고, 공정은 우선작업순위(precedence)로 정의될 수 있다. 이 외에도, 긴급주문, 동시작업, 연속작업, 고정된 스케줄(fixed schedule), 요일별 조건이나 기준들도 포함된다.

이 연구에서는, 첫째로, 일반화된 job-shop 모형의 제약조건들을 표현하고, 제약조건을 만

족시키는 일정계획 수립방법을 제시한다. 우리는 이것을 “제약일정계획”이라 한다(서민수,1995 ; 정남기,1995). 제약일정계획은 흔히 제약만족문제(constraint satisfaction problem) 모형을 이용한다. 이 일정계획은 디스패칭 방식으로 생성된 것과는 달리, 미리 정해놓은 기준을 만족시키는 것을 목표로 한다. 납기를 예로 들면, 제약일정계획은 납기만족이라는 목표가 미리 정해지지만, 디스패칭 방식은 일정계획 후 납기 만족여부를 확인해보는데 그친다.

둘째로, 이 연구는 제약조건 중에서 불확정 제약조건을 파악하여 표현하고, 이를 효과적으로 추론하는 방법을 강구한다. 납기, 작업시간, 주문의 불확정성을 표현한 후, 이 상황을 고려한 일정계획 방법을 정한다. 불확정 제약조건 추론의 결과, 적절한 일정계획 방법이 선정되면, 일정계획은 제약식의 전파(propagation)에 의해 추론된다. 즉, 납기, 각 공정의 시작시간과 완료 시간으로부터 한 공정의 일정계획이 정해지면, 이것은 새로운 제약식으로 추가되어 다른 공정과 기계에 전파된다.

이 연구에서는 소프트웨어의 再使用성과 일정계획 적용범위의 확장성을 높이기 위하여, OOP(Object Oriented Programming: 客體 指向 프로그래밍)기법을 사용하고자 한다. 즉, 문제 표현, 제약, 탐색 등을 각각의 모듈별로 정의하고 각각의 모듈들이 주변 상황에 따라 대응할 수 있도록 한다. 이 시스템은 workstation에 설치되어 운영되게 한다. UNIX 운영체제와 C++ 언어를 사용하고, 사용자의 편의를 위한 입출력 도구들도 설치한다.

이 논문의 구성은 다음과 같다. 다음 2절에서는, 일반화된 job-shop 모형을 구체적으로 소개하며, 이에 연관된 불확정제약조건을 설명한다. 3절은 불확정제약조건의 추론과정을 일정계획

모듈과 함께 설명한다. 4절은 연구결과로서, 일정계획 자료의 입력방법과 출력내용을 소개한다.

## 2. 연구모형

### 2.1 일반화된 job-shop모형

job-shop의 기본 정의를 포함하고, 그 외에 다른 특성들을 추가하여 모형을 세운다. 이 모형은 자동 생산 환경(FMS)이나 흐름 공정(flow-shop)의 일정계획, 프로젝트 일정계획, 또는 설비 수리 계획 등에 적용 가능하다. 이 논문에서 사용하는 가정들은,

- ① 작업 준비 시간은 작업시간에 포함된다.
- ② 각 공정은 하나의 자원을 사용한다
- ③ 각 작업 마다 도착 시간과 납기가 정해져 있다.
- ④ 기본적인 제약, 즉, 납기, 부품 제약, 자원 제약을 고려한다.
- ⑤ 선점(preemption)은 없다.

이 논문에서 사용하는 기호를 다음과 같이 정의해 둔다.

- $A_{ij}$  : i번째 주문의 j번째 공정
- $PT(A_{ij})$  : i번째 주문, j번째 공정의 가공시간 (Processing Time)
- $EST(A_{ij})$  : i번째 주문, j번째 공정의 가장 이른 작업 시작시간 (Earliest Start Time)
- $LFT(A_{ij})$  : i번째 주문, j번째 공정의 가장 늦은 작업 종료시간 (Latest Finish Time)

#### (1) 도착 시간

주문이 도착 할 때 필요한 공정 순서가 정해

진다. 주문의 도착 시간은 첫 공정의 EST로 지정된다.

#### (2) 납기

주문을 완료하여 고객에 납품하는 시기이다. 납기만족은 일정계획에서 중요한 기준이다. 납기는 마지막 공정의 LFT로 지정된다.

#### (3) 부품 제약

하나의 부품은 동시에 여러 자원에서 작업되어질 수 없다. 각 부품은 정해진 자원에서만 작업이 이루어지며, 동시에 다른 자원에서 작업되어질 수 없다. 즉, 공정  $A_{ij}$ 가 공정  $A_{jk}$ 에 선행한다면 다음 식을 만족해야만 한다.

$$EST(A_{ij}) + PT(A_{ij}) \leq EST(A_{jk})$$

$$LFT(A_{jk}) - PT(A_{jk}) \leq LFT(A_{ij})$$

#### (4) 자원 제약

하나의 자원에서 동시에 두 개의 작업이 이루어질 수 없다. 즉, 공정  $A_{ij}$ 가 공정  $A_{kl}$ 을 선행할 때, 다음 식이 만족해야 한다.

$$EST(A_{ij}) + PT(A_{ij}) \leq EST(A_{kl})$$

$$LFT(A_{kl}) - PT(A_{kl}) \leq LFT(A_{ij})$$

#### (5) 순서제약

두개의 서로 다른 작업  $A_{ij}$ 와  $A_{kl}$ 이, 작업의 특성상 순서대로 진행될 필요가 있을 때, 이 제약을 지정한다. 이 같은 경우는, 작업준비 시간이나 작업 흐름상의 문제일 수도 있다. 순서제약은, ① 순서만 지정하는 경우와 ② 선행작업이 끝나는 시간이 후행작업의 시작 시간이 되는 연속작업의 경우로 나눌 수 있다. 각각의 제약 조건을 수식으로 표현하면 다음과 같다. 단, 조정되는 시구간(時區間)의 값들은 처음의 시구간을 벗어나서는 안된다.

①순서만 지정하는 경우 :  $A_{ij}$ 가  $A_{kl}$ 에 선행

$$EST(A_{kl}) = \text{Max}[EST(A_{kl}), EST(A_{ij}) + PT(A_{ij})]$$

$$LFT(A_{ij}) = \text{Min}[LFT(A_{ij}), LFT(A_{kl}) - PT(A_{kl})]$$

②연속 작업의 경우 :  $A_{ij}$ 와  $A_{kl}$ 은 연속

$$EST(A_{ij}) = \text{Max}[EST(A_{ij}), EST(A_{kl}) - PT(A_{ij})]$$

$$LFT(A_{ij}) = \text{Min}[LFT(A_{ij}), LFT(A_{kl}) - PT(A_{kl})]$$

$$EST(A_{kl}) = \text{Max}[EST(A_{kl}), EST(A_{ij}) + PT(A_{ij})]$$

$$LFT(A_{kl}) = \text{Min}[LFT(A_{kl}), LFT(A_{ij}) + PT(A_{kl})]$$

#### (6) 동시작업

하나의 작업  $A_{ij}$ 가 작업을 시작하면 또 다른 작업  $A_{kl}$ 이 동시에 시작되거나,  $A_{ij}$ 가 작업을 끝냈을 때  $A_{kl}$ 도 동시에 끝나도록 하는 제약을 말한다. 두개의 작업  $A_{ij}$ 와  $A_{kl}$ 의 작업시간이 같은 경우, 동시에 시작하고 동시에 끝나는 제약도 있다. 각각의 제약 조건은 다음과 같다.

① 동시에 시작

$$EST(A_{ij}) = \text{Max}[EST(A_{ij}), EST(A_{kl})] = EST(A_{kl})$$

② 동시에 종료

$$LFT(A_{ij}) = \text{Min}[LFT(A_{ij}), LFT(A_{kl})] = LFT(A_{kl})$$

③ 동시에 시작하고 동시에 종료

$$EST(A_{ij}) = \text{Max}[EST(A_{ij}), EST(A_{kl})] = EST(A_{kl})$$

$$LFT(A_{ij}) = \text{Min}[LFT(A_{ij}), LFT(A_{kl})] = LFT(A_{kl})$$

#### (7) 동시작업 회피

동시작업과 반대로, 두개의 작업이 동시에 작업되어서는 안되는 경우의 제약 조건이다.

$$EST(A_{ij}) + PT(A_{ij}) \leq EST(A_{kl})$$

$$LFT(A_{kl}) - PT(A_{kl}) \geq LFT(A_{ij})$$

또는

$$EST(A_{kl}) + PT(A_{kl}) \leq EST(A_{ij})$$

$$LFT(A_{ij}) - PT(A_{ij}) \geq LFT(A_{kl})$$

#### (8) Calendar Work Period(CWP)

하나의 일정계획은, 시간의 흐름을 연속적인 것으로 생각한다. 그러나, 실제 작업 환경에서 시간의 흐름을 단속적으로 고려하여야 할 경우가 발생할 수 있다. 어떤 작업에 대하여, 실제 달력상의 일정을 고려하여 일정계획을 작성하고 싶은 경우, 이에 대한 제약을 추가할 수 있다. 예를 들어, 토, 일요일의 작업을 회피하고 싶은 어떤 작업  $A_{ij}$ 가  $PT(A_{ij})=2$  일이고, 이 작업의 작업 가능 시구간이 실제 달력상에서 [화, 수, 목, 금, 토, 일, 월]이라 하면, 이 작업의 작업 가능 시구간에서 [토, 일, 월]은 제외된다. 이 제약은 설비수리 일정계획 문제나 프로젝트 일정계획 등에 효과적으로 사용되어 질 수 있다.

## 2.2 불확정 제약 조건

초기에 생성된 스케줄을 가지고 작업을 진행하다 보면 주문, 자원(기계, 작업자), 공정 가공시간의 변화가 발생한다. 또, 도중에 일정계획 목표가 변경될 수도 있다. 작업을 시작하기 전에 이 변화를 최대한 예상하더라도, 예기치 못한 상황들은 여러 가지 수정을 불가피하게 한다.

우리는 이 연구에서 이러한 예기치 못한 상황의 발생들을 불확정 제약조건(uncertain constraints)으로 정의한다. 주문과 공정의 데이터를 변화시키는 요인들은, 자원의 고장과 할당 변경, 공정 작업시간의 변화, 새로운 주문의 도착 또는 주문취소, 일정계획 목표의 변경 등 다양하다.

#### (1) 자원의 고장

생산 자원이 항상 완전할 수는 없다. 작업자의 오작동이나 무리한 작업, 자원의 노후화 등으로 인해 자원이 고장 날 수 있다. 또, 생산 환경의 변화에 따라서 자원이 추가되거나, 없어질 수도 있으며, 기존의 자원보다 효율이 뛰어난 자원으로 대체될 수도 있다. 이 논문에서는, 자원의 고장에 대해서만 고려한다.

#### (2) 공정 작업시간의 변화

작업시간은 예상 시간보다 짧아질 수도, 길어질 수도 있다. 작업 시간은 기대치이므로 기대치 보다 빨리, 또는 늦게 작업될 수 있다. 또, 실제 상황에서 생산 자원은 작업시간을 정할 때의 자원 효율을 항상 일정하게 유지하지 못하기 때문에 원래 입력된 문제를 그대로 따르지 못한다. 또는, 작업자의 작업 숙련도나 태도에 따라서 달라질 수 있다. 이러한 변화는 원래 일정계획에서 시작 가능 시간을 늦추거나 앞당기게 된다.

#### (3) 새 주문의 도착

기존의 주문이 계획되어 있는 상황에서 새 주문이 도착하면, 일정계획 수정이 불가피하다. 새 주문에 대한 우선 순위를 결정하고, 일정계획을 조정해야 한다.

#### (4) 일정계획 목표의 변경

관리자가 추구하고자 하는 목표들은 다양하게 존재한다. 예를 들면, 주문의 납기 만족을 추구할 수도 있고, 가능하면 지연되는 주문의 수가 적은 것을 바라기도 한다. 또, 사용되고 있는 자원의 효율을 높이기 위하여 총처리시간(makespan)이나 평균체재시간(mean flowtime)의 최소화를 우선적 목표로 선택할 수 있다.

### 3. 불확정 제약조건의 추론

앞 절에서 언급한 4가지의 불확정 제약조건들은 여러가지 조합으로 발생된다. 이제 제시하려는 것은 이들 여러가지 경우에 합당한 일정계획 방법들이다.

#### 3.1 추론을 위한 모듈화

먼저, 이 모든 불확정 제약조건들을 처리하기 위해서는 일정계획 각 기능들의 모듈화가 필요하다. 즉, 자료 입력기능, 초기 일정계획 생성기능, 일정계획 변경기능, 출력기능 등을 독립적 모듈로 구성하고, 불확정 제약조건에 따라 필요한 기능을 선택하여 사용하게 한다. 이것은 기능의 중복을 피할뿐 아니라, 앞으로 기능의 확장이나 변경을 쉽게 할 수 있다는 이점이 있다. 이 연구에서 구현한 모듈은 다음과 같다.

##### (1) 입력 모듈

주문, 자원과 일반화된 job-shop에서 고려하는 모든 제약조건들에 관한 자료를 입력시켜 파일로서 보관한다. 또 불확정 제약조건의 변동상황을 수시로 입력시킬 수 있도록 사용자 인터페이스를 갖고 있다.

##### (2) 출력 모듈

일정계획의 결과를 간트차트와 수치로서 출력시킨다.

##### (3) 제약전파 모듈

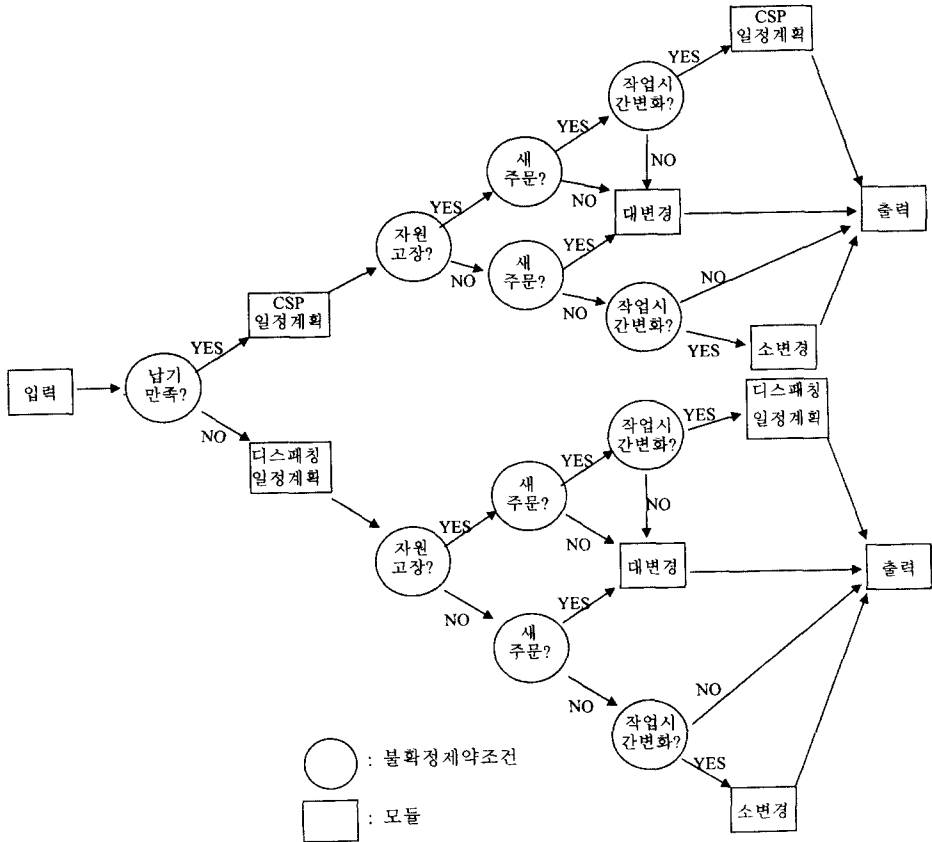
일정계획 생성과정이나 일정계획에 변화가 발생하는 경우에, 일정계획이 확정된 공정의 시작 시간과 완료시간을 아직 확정되지 않은 공정에

서 감안하게 하여, 그곳에서 각종 제약조건(기계제약, 부품제약, 순서제약 등)을 어기지 않도록, 미확정 공정들의 EST와 LFT를 조정한다.

이것은 다른 모듈들의 하위모듈로서, 다음에 설명할 CSP일정계획 모듈, 디스패칭 일정계획 모듈, 대변경 모듈, 소변경 모듈에서 공통으로 사용된다.

(4) CSP 일정계획 모듈

제약만족문제(Constraint Satisfaction Problem) 모형을 이용하여, 납기만족을 위한 일정계획을 생성시킨다. 이 모형에서는 시구간 즉, EST와 LFT의 구간을 대상으로 납기를 만족시키는 가능한 스케줄을 탐색한다.



[그림1] 불확정 제약조건의 추론과정

(5) 디스패칭 일정계획 모듈

디스패칭 규칙을 이용하여 일정계획을 생성시킨다. 예를 들어, 평균 체재 시간을 줄이기 위해서는 SPT(shortest processing time) 규칙을, 총처리 시간을 줄이기 위해서는 LWKR(least work

remaining) 규칙을 적용한다.

(6) 소(小)변경 모듈

어느 한 공정의 작업시간에 변화가 있을 때, 이에 영향받는 다른 공정의 일정계획을 조정한다.

다. 즉, 자원의 제약과 주문의 순서 제약을 어기지 않는 범위에서 변화된 시간만큼 왼쪽 또는 오른쪽으로 일정계획을 옮긴다.

(7) 대(大)변경 모듈

자원이 고장 나거나 주문의 변경이 있을 때, 변경시점 이전의 일정계획은 건드리지 않으나, 그 이후 계획을 전부 재생성하기 위한 모듈이다. 대변경의 대상(CSP 일정계획 또는 디스패칭 일정계획)에 맞도록, 변경된 사항을 갱신하여 변경시점 이후의 일정계획을 생성시킨다.

이 모듈들 중, (1)~(3)은 어느 불확정 제약조건에서나 항상 공통으로 사용되는 모듈들이고, 나머지들은 경우에 따라 선택적으로 사용된다.

3.2 추론과정

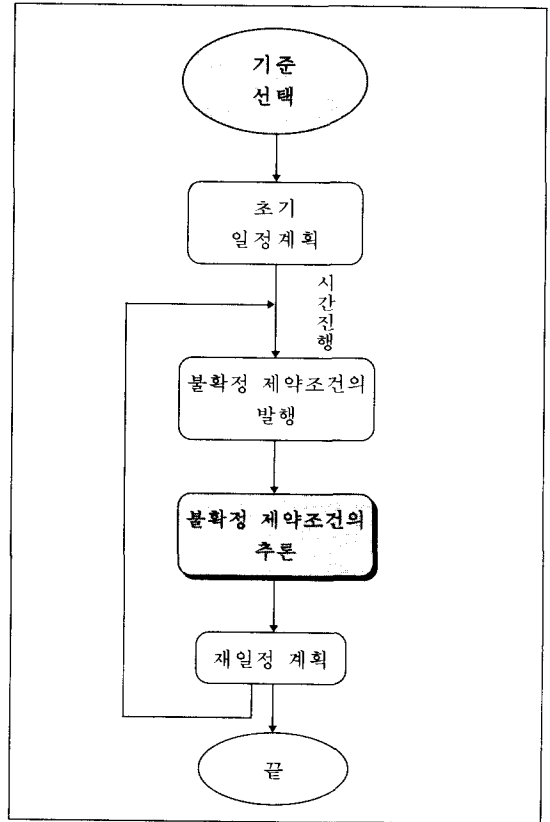
이제 불확정 제약조건에 따라 이 모듈들을 선택적으로 사용하기 위한 추론과정을 살펴보자. 이 추론과정은 위 [그림1]과 같이 표현된다.

먼저, 일정계획의 목표에 따라, 납기만족이 우선이면 CSP 일정계획 모듈을 사용하고, 그렇지 않으면 디스패칭 일정계획 모듈을 사용한다. 그 다음 자원의 변경, 주문의 변경, 공정 작업시간의 변경에 따라 선택 모듈을 달리한다.

새 주문 및/또는 자원 고장이 생기면, 그 시점까지의 스케줄은 고정시키고, 그 이후 스케줄은 다시 생성한다(대변경). 이때, 새 주문은 다른 공정에 주는 영향이 크므로, 만약 기존 주문의 작업시간도 바뀌었으면, 기존의 다른 공정들을 포함해서 아예 다시 일정계획을 수립한다.

작업시간만 변한 경우는 이보다 간단하다. 앞의 소변경 모듈에서 설명한 바와 같이, 어느 한 공정의 작업시간 변화에 영향 받는 다른 공정의 일정계획을, 자원 제약과 순서제약을 어기지 않

는 범위에서, 변화된 시간만큼 왼쪽 또는 오른쪽으로 옮긴다.

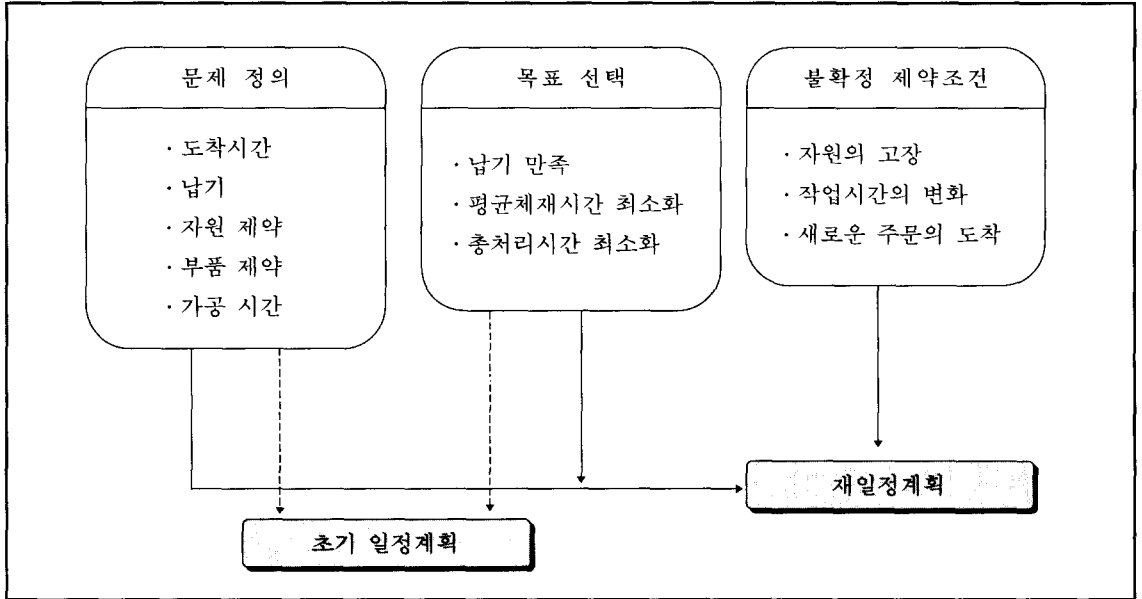


[그림2] 일정계획의 흐름도

4. 연구결과

이 연구에서 개발한 일정계획 시스템의 전체적인 과정은 [그림 2]와 같다. 먼저, 현재의 상황과 자료를 사용하여 초기 일정계획이 생성된다.

초기 일정계획은 관리자의 판단 기준, 즉, "납기를 만족시키는 일정계획을 구할 것인가? 또는, 공정 내에서 작업의 평균 체재시간을 중심으로 일정계획을 구할 것인가?"에 따라 달라진



[그림3] 초기 일정계획과 재일정계획

다. 그리고, 주어진 일정계획을 기준으로 상황의 변화, 즉 불확정 제약조건들을 반영시키면서 일정계획이 조정된다. 이 재일정계획은 불확정 제약조건이 추론되어 적절한 일정계획 모듈이 선택됨으로써 실현되며, 대변경과 소변경을 포함하는 개념이다.

다음 [그림3]은 초기 일정계획과 재일정계획을 사용되는 자료(지식)의 관점에서 정리한 것이다. 재일정계획은 초기 일정계획에 비해 불확정 제약조건을 추가하여 고려하고 있음을 보인다. 더 많은 다양한 변화에 대처하기 위해서는, 더 많은 변화에 대한 자료(지식) 추가가 요청되며, 이에 합당한 모듈들이 준비되어야 한다.

#### 4.1 데이터 입력용 언어

여기서 다루는 일반화된 job-shop의 입력자료는 다음 2가지로 구성된다. 즉 자원의 형태, 주

문, 공정, 일반 제약조건은 데이터 파일을 구성하여 입력시키고, 불확정 제약조건은 발생될 때마다 화면에서 직접 입력시킨다.

데이터 파일을 구성하는 일은 문제가 커짐에 따라 자료들을 서로 구별시키기가 어려워지므로, 점점 더 힘들어진다. 즉, 자원, 공정, 주문에 관한 자료들을 모두 숫자로만 표현하면, 이들 사이의 구별이 어렵게 된다. 이 혼동을 줄이기 위해, 여기서는 입력을 위한 언어를 고안하였다. 자원, 공정, 주문에 관한 자료들을 보통의 언어로 표현하고 이를 프로그램이 해독하게 함으로써, 나중에 이 자료들을 수정하거나 참조할 때 혼동을 줄이고자 하였다. 다음은 이 언어의 사용 방법을 보인 것이다.

##### (1) 자원에 관한 단어

자원을 자원클래스, 자원타입, 단위자원의 3단계로 구분하고, 하위 단계는 상위단계의 속성을



계승한다.

```
resource__class {자원}
resource__type {자원타입, 자원}
resource__pool {단위자원, 자원타입}
```

(2) 주문에 관한 단어

```
task {주문}
earliest__start {주문, 도착시간}
lateset__end {주문, 납기}
```

(3) 공정에 관한 단어

```
operation {공정, 주문명}
before {선행공정, 후행공정}
duration {공정, 가공시간}
resource__request {공정, 자원타입}
```

(4) 제약조건에 관한 단어

fixed {공정, EST, LFT} : 일정이 이미 고정된 공정의 시구간을 표현.

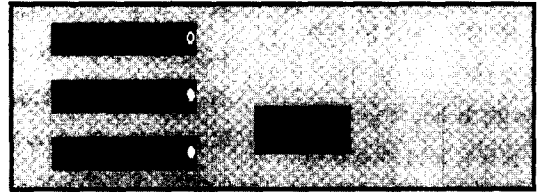
succeed\_\_continue {공정1,공정2} : 연속작업인 두 공정을 표현.

sync\_[start, end, startend] {공정1, 공정2} : 두 공정이 동시에 시작, 종료, 시작 및 종료될 조건.

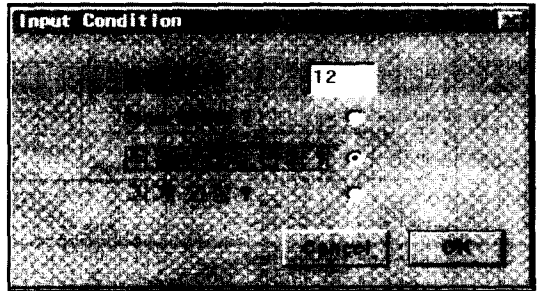
non\_sync {공정1, 공정2} : 두 공정간의 동시 작업 회피 제약조건.

calendar {공정} : 주말작업 회피 제약조건.

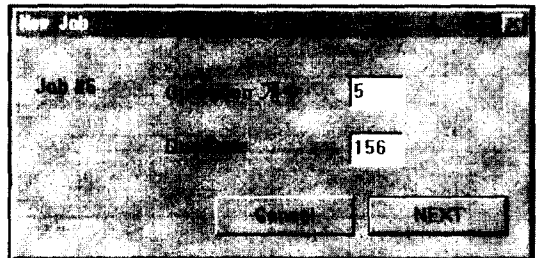
불확정 제약조건들은 발생 때마다 화면에서 직접 입력된다. 예를 들면, [그림4]는 목표를 선택할 수 있는 화면이며, [그림5]는 자원, 주문, 작업시간의 변화를 입력시키는 화면이다. [그림6]과 [그림7]은 새 주문의 내용을 구체적으로 입력시키는 화면이다.



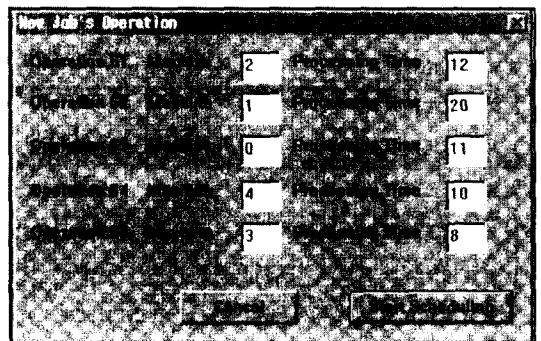
[그림 4] 목표 선택 화면



[그림 5] 자원, 주문, 작업시간 변화를 입력시키는 화면



[그림 6] 새 주문의 공정과 납기 입력 화면

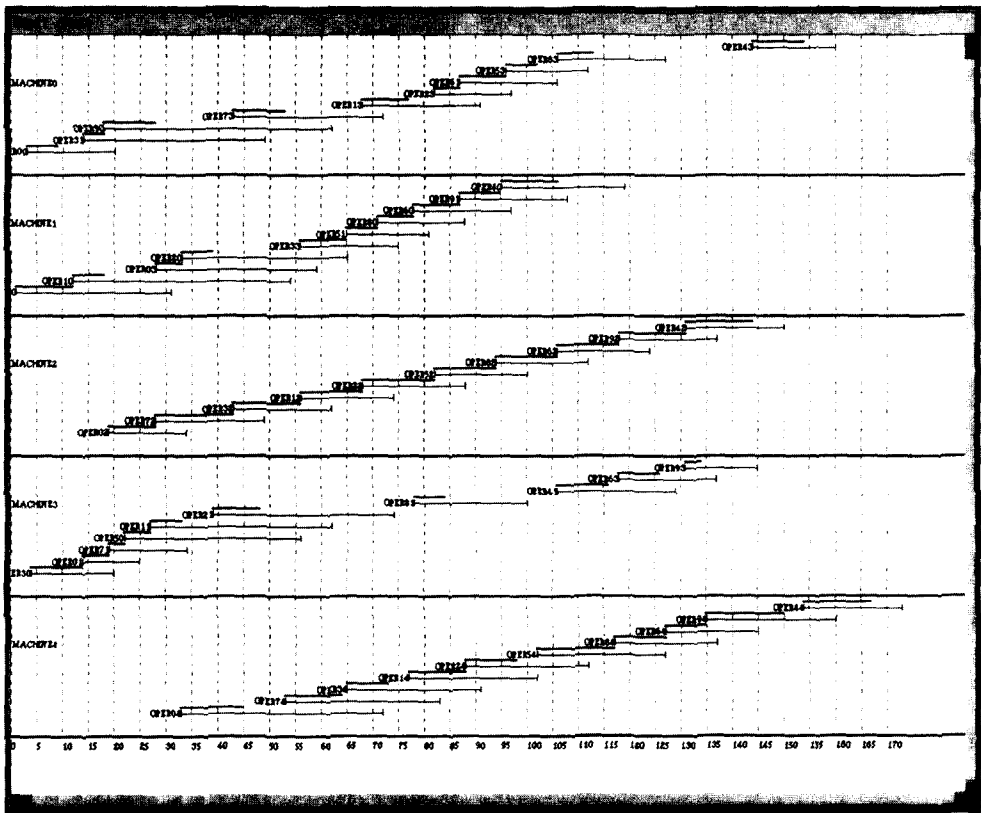


[그림 7]

### 4.2 일정계획 결과 출력

일정계획 결과는 간트(gantt) 차트 또는 문서로 출력된다. [그림8]의 간트 차트에서 굵은선(——)은 (EST, EST+PT)를 나타내고, 가는선(——)은 (EST, LFT)의 시구간을 나타낸다. 즉, 굵

은선은 각 공정이 납기를 제외한 다른 제약조건을 만족 시키면서 실제 작업될 일정을 나타내고, 가는선은 납기를 포함한 모든 제약 조건을 만족시키기 위해서 공정이 작업되어야할 시구간을 나타낸다. 그러므로, 만약 한 주문에 대한 모든 공정에서 굵은선의 구간이 가는선 구간 안



[그림 8] 일정계획 결과의 간트 차트 출력

에 포함되면, 이 주문의 납기가 지켜지는 일정 계획이 되나, 그렇지 못하면 납기를 지킬 수 없는 일정계획이 된다.

이 시스템에서는, 관리자의 일정계획 목표에 따라 납기만족의 의미가 달라지므로, 만약 총처리시간이나 평균체재 시간에 관심이 많은 관리

자라면, 굵은선의 일정계획을 보면 될 것이다

[그림 9]와 같은 문서 출력은, 출력 자료를 다른 프로그램에서 활용할 때 유익할 것이다. 이 자료를 자동으로 입력시켜 능력계획이나 작업 지시계획에 사용할 수 있다.

이 출력 자료들은 대 변경이나 소 변경의 재

일정계획에서도 출력된다. 자원, 주문, 공정의 변화들이 발생하면 새로운 사건들을 입력 받고, 이에 대해서 재일정계획을 수행하여 이 자료들을 출력시킬 수 있다.

파일(F)	편집(E)	찾기(S)	도움말(H)
<b>MACHINE0</b>			
Operation	EST	LFT	Processing Time
OPER00	3	20	6
OPER31	14	49	4
OPER90	10	62	10
OPER73	43	72	10
OPER13	68	91	9
OPER23	82	97	5
OPER41	87	106	9
OPER53	96	112	6
OPER83	106	127	7
OPER43	144	160	10
<b>MACHINE1</b>			
Operation	EST	LFT	Processing Time
OPER70	1	31	11
OPER10	12	54	6
OPER03	28	59	5
OPER20	33	65	6
OPER33	56	75	9
OPER51	65	81	6
OPER00	71	88	7
OPER60	78	97	9
OPER91	87	100	8
OPER40	95	119	11
<b>MACHINE2</b>			
Operation	EST	LFT	Processing Time
OPER02	19	34	9
OPER72	20	49	15
OPER32	43	62	13
OPER12	56	74	12
OPER22	68	88	14
OPER52	82	100	12
OPER02	94	112	12

[그림 9] 일정계획 결과의 문서 출력

## 5. 결 론

이 연구에서 개발된 일정계획 시스템은, 여러 가지 확장된 제약조건이 있는 job-shop에서, 원하는 여러 가지 목표를 선택할 수 있게 하고, 각 목표의 특성에 따라 일정계획을 생성한다.

그리고, 동적인 상황에서 예기치 않은 사건이 일어날 때, 이 사건들의 상황에 따른 각각의 특성을 반영하여 재일정계획을 실시한다.

이 결과들은, (1)일정계획에 필요한 기능들을 여러개의 모듈로 분할하고, (2)관련된 제약조건들을 일반적인 제약조건과 불확정 제약조건으로 구분하고, (3)불확정 제약조건을 추론하여 적절한 일정계획 모듈을 선택함으로써 실현되었다. 또, (4)사용자의 편의를 위하여, 문제 입력을 위한 언어를 작성하였으며, 출력 결과를 간트 차트로 표현하였다.

이 논문의 의의는, 이런 방법과 과정을 통해 지금까지 연구된 발견적인 디스패칭 규칙과 제약만족 모형을 하나의 틀에서 함께 이용함으로써, 여러가지 변화에 대처하여 일정계획을 수립할 수 있는 시스템을 개발하였다는데 있다.

앞으로, 여기에서 제시된 방법을 확장하여, 보다 광범위한 문제를 다룰 수 있도록 여러 가지 모듈들을 추가하는 작업이 필요할 것이다. 특히 디스패칭 규칙들을 보완하거나, 상황의 변화 요인을 추가하는 일이 요청된다.

## 참 고 문 헌

1. 기병성, 박재성, 정남기, "動的 상황을 위한 規則기반 制約 스케줄링 시스템", 대한 산업 공학회 95 추계 학술대회 발표논문집, 경희대학교, 1995.
2. 서민수, 김재연, 최해운, 신동익, "제약조건 추론을 통한 다단계 일정계획 수립 방법론", 한국전문가시스템학회 95 춘계 학술대회 발표 논문집, (1995), 87~93.
3. 정남기, "分散管理 시스템을 위한 動的 스케줄링의 연구", 대한산업공학회지, 21권 2호 (1995), 207~216.
4. Beck, H., *The Management of Job-shop Scheduling Constraints in TOSCA*, AIAI-TR-121, University of Edinburgh, United

- Kingdom, 1993.
5. Emilio, P., P. Stefano. and Giuseppina, *Scheduling in S.O.S.*, Department of Eletronic, Polytechnic of Milan, Milano, Italy.
  6. Fox, M. S., J. F. Chionglo and M. Barbuceanu, *The integrated supply management system*, Working Paper, University of Toronto, Ontario, Canada, 1993.
  7. Keng,N.and D.Y.Yun. *A planning/scheduling methodology for constrained resource problem*, In proc. IJCCAI-89, Detroit, MI, 1989.
  8. Ow. P.S., and S. F. Smith, *Viewing scheduling as an opportunistic problem-solving process*, Working Paper, Carnegie Mellon University, Pittsburgh, PA, 1986.
  9. Tsukiyama, M., K. Mori. and T. Fukuda, *Dynamic Scheduling using Scheduling Editor and Distributed Decision Maker*, IFAC-INCOM '92 (1992), 84~89.