

## 조선의 일정계획 수정 및 통제: DAS-REACT

이정승\*, 이재규\*, 최형림\*\*

### Reactive Scheduling and Control for Shipbuilding : DAS-REACT

Jung Seung Lee\*, Jae Kyu Lee\* and Hyung Rim Choi\*\*

#### 요 약

전체적인 조선 생산일정계획을 수립하기 위해서 계층적 구조를 채택하였다. 이 구조에 의하면 도크의 상위 일정계획 생성기는 탑재 일정계획을 생성하고, 조립공장의 하위 일정계획 생성기는 상위 일정계획 생성기가 요구하는 범위 내에서 각각 세부적인 조립 일정계획을 세우게 된다. 그런데 하위 일정계획 생성기가 아무리 노력하여도 상위 일정계획 생성기로부터의 제약조건을 만족시킬 수 없는 상황이 발생하면, 일정계획 수정기에 통보하여 초기 탑재 일정계획을 수정해야 한다.

하위 일정계획 생성기로부터의 조정요청을 반영하기 위해서 부분 수정을 통해 충격을 최소화시키는 일련의 휴리스틱으로 구성된 알고리즘을 제시하였다. 각각의 휴리스틱은 여유시간 제거, 버퍼 제거, 작업시간 축소, 다단계 여유시간 제거, 과부하 자원 공유블럭 조정, 자원 제약조건 완화 등이고, 흡수되지 않은 일부의 충격은 주변으로 전파된다. 이와 같은 방법론을 대우조선의 상황에 적용하여, 조선 일정계획 수정 시스템인 DAS-REACT를 개발하였다.

Key Words : 일정계획 수정(Reactive Scheduling), 일정계획 통제 (Reactive Control), 일정계획 수립(Scheduling)

\* 한국과학기술원 테크노경영대학원

\*\* 동아대학교 경영정보학과

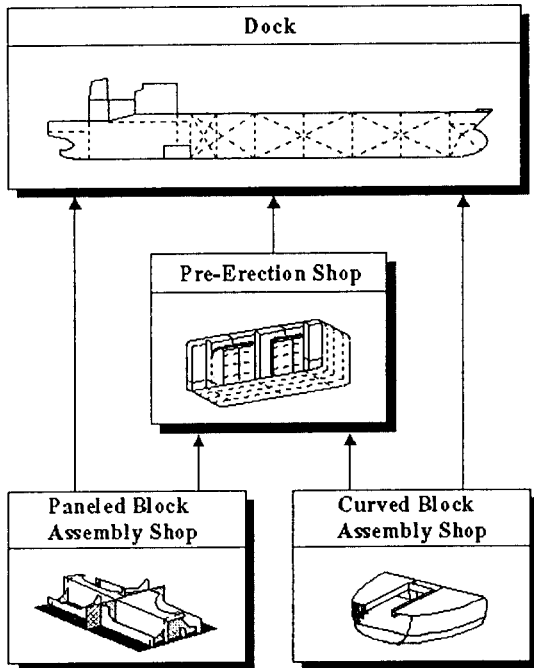
# 1. 서론

## 1.1. 연구의 배경

### 1) 전형적인 선박건조 과정

선박은 어떻게 건조되는가? 거대한 선박을 도크(dock)에서 한번에 생산한다면 생산성이 매우 떨어질 수밖에 없다. 따라서 선박 설계 시 인력과 시설이 허용하는 범위 내에서 작업이 용이 하도록 <그림 1>처럼 거대한 선박을 적절한 크기의 '블럭(block)'으로 분할하게 되는데, 대개 한 척의 선박을 건조하기 위해서는 400~500개의 블럭이 필요하다.

이러한 블럭은 각 블럭의 외형에 따라 크게 밑바닥이 평평한 '평블럭(Paneled Block)'과 밑바닥이 곡면인 '곡블럭(Curved Block)'으로 나눌 수 있다.



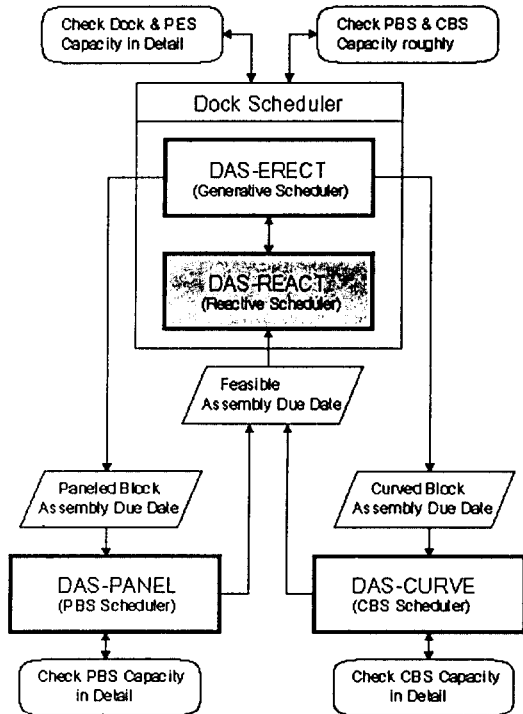
<그림 1> 전형적인 선박건조 과정

특성이 다른 두 종류의 블럭을 조립하는데 필요한 시설이 서로 다르기 때문에, 평블럭은 '평블럭 조립 공장(Paneled Block Assembly Shop)'에서 곡블럭은 '곡블럭 조립공장(Curved Block Assembly Shop)'에서 생산된다. 해당 조립공장에서 생산된 블럭은 도크로 옮겨져 일정한 순서에 따라 거대한 크레인(crane)으로 하나씩 쌓아올리는 과정을 통해 선박의 형태를 만들게 되는데, 이러한 과정을 '탑재(erection)'라고 한다. 또한 도크의 생산성을 더욱 높이기 위해 일부 블럭의 경우 도크에서 바로 탑재하지 않고, 도크 주변의 '선행탑재장(Pre-Erection Shop, PE Shop)'에서 몇 개의 블럭을 미리 결합하여 '슈퍼블럭(Super Block)'으로 만든 후 탑재하기도 한다.

### 2) 계층적 구조에서의 조선 일정계획 수립 과정

이와 같은 전형적인 선박 건조과정에 비추어 볼 때, 전체적인 조선 일정계획은 '자원제약하의 계층적 일정계획 수립문제(Hierarchical Scheduling under Resource Constraints)'로 볼 수 있다. 이러한 계층적 구조에 의하여 <그림 2>처럼 상위 일정계획 생성기(higher-level scheduler)는 도크와 선행탑재장의 탑재 일정계획을, 하위 일정계획 생성기(lower-level scheduler)는 다수의 블럭 조립공장의 조립 일정계획을 생성한다.

조선 일정계획 수립과정을 살펴보면 다음과 같다. '일정계획 생성단계(Generative Scheduling Stage)'에서 상위 일정계획 생성기는 도크와 선행탑재장의 생산능력을 상세히 고려하고, 조립 공장의 생산능력을 개략적으로 고려하여 초기 탑재 일정계획을 생성한다. 동시에 그 결과로 생성된 각 블럭의 조립완료일을 해당 조립공장에 넘겨준다. 그러면 하위 일정계획 생성기는 해당 조립공장의 생산능력을 상세히 고려하면서 조립완료일을 만족시키는 조립 일정계획을 생성한다. 그런데 하위 일정계획 생성기가 아무리



〈그림 2〉 계층적 구조에서의 조선 일정계획 수립 과정

노력하여도 상위 일정계획 생성기로부터 주어진 조립완료일을 만족시킬 수 없는 블럭이 발생할 수 있다. 이때 주어진 조립완료일을 만족시킬 수 없는 블럭을 가리켜 '실패블럭(failed block)'이라 명하고, 그 실패블럭의 초기 탑재 일정계획을 변경해야 한다.

### 1.2. 연구의 목적 및 범위

조선의 계층적 일정계획 수립문제에서는 상위 일정계획 생성기와 하위 일정계획 생성기 사이에 일정계획의 불일치가 발생할 수밖에 없고, 조선 일정계획을 완성하기 위해서는 이를 반드시 제거해야만 된다. 이러한 상황을 해결하기 위해 도입된 '일정계획 수정단계(Reactive Scheduling Stage)'에서는 탑재 일정계획을 다시 생성하는(rescheduling) 대신, 하위 일정계획 생성기로부터 실패블럭의 새로운 조립완

료일을 일정계획 수정기(reactive scheduler)에 전달하여 초기 탑재 일정계획을 부분적으로 수정한다.

본 논문의 목적은 조선 일정계획을 수정하기 위한 방법론을 설계하고, 조선 일정계획 수정시스템을 구현하는데 있다.

한편, 본 연구는 1991년 1월부터 현재까지 한국 과학기술원 지능정보시스템 연구실과 대우중공업이 공동으로 진행중인 '대우조선의 일정관리 전문가 시스템의 개발 : DAS Project'의 일환이다.

본 논문의 구성을 간략히 설명하면, 이어지는 2절에서는 본 연구와 관련 있는 기존의 문헌들을 간략하게 고찰하고, 3절에서는 조선 일정계획 수정을 위한 방법론을 제시하고, 4절에서는 일정계획 수정 시스템, DAS-REACT를 소개한 후 마지막으로 5절에서는 결론을 제시하였다.

## 2. 문헌 조사

### 2.1. 일정계획 생성, 수정, 통제와 인공지능 접근 방법

일정계획 수립문제는 전형적인 NP-hard 문제로 과거 수없이 많은 연구가 진행되어 왔고, 또 현재에도 진행되고 있다. 일정계획 수립문제를 해결하기 위한 접근방법으로는 크게 OR 기법 등을 사용하는 '전통적인 접근방법(Conventional Approach)'과 최근 각광 받고 있는 인공지능기법을 이용한 '인공지능 접근방법(AI-Based Approach)'이 있다.

일반적으로 일정계획 수립문제에 있어 인공지능 접근방법은 계산상의 부담(combinatorial complexity)을 덜어 주고, 학습(learning)이 가능하며, 일정계획을 동적으로 재수립 또는 수정(dynamic rescheduling or reactive scheduling)이 가능하고, 표현력이

방법론 \ 접근 방법	전통적인 접근방법	인공지능 접근방법
문제 해결 방법	<ul style="list-style-type: none"> <li>· Mathematical Programming</li> <li>· Simulation</li> <li>· Heuristics</li> </ul>	<ul style="list-style-type: none"> <li>· Constraint Satisfaction Problem</li> <li>· Constraint Directed Search</li> <li>· Rule-Based Scheduling</li> <li>· Heuristic Search</li> <li>· Case-Based Reasoning</li> <li>· Artificial Neural Network</li> <li>· Fuzzy Logic</li> <li>· Genetic Algorithm</li> </ul>
Architecture		<ul style="list-style-type: none"> <li>· Centralized</li> <li>· Distributed (including Blackboard)</li> <li>· Hierarchical</li> </ul>

〈표 1〉 일정계획 수립문제에 대한 전통적인 접근방법과 인공지능 접근방법

뛰어나 전통적인 접근방법보다 우수하다고 알려져 있다. 따라서 이미 생성된 일정계획을 수정, 통제(reactive scheduling and control)하기 위해서는 인공지능 접근방법이 적합하리라 여겨진다.

일정계획 수립문제에 대한 전통적인 접근방법과 인공지능 접근방법의 구체적인 방법론은 다음 〈표 1〉과 같다.

## 2.2. 일정계획 생성, 수정, 통제와 Architecture

일정계획 수립에 관한 연구는 일정계획 생성만 다룬 연구와 일정계획 생성뿐 아니라 수정, 통제를 함께 다룬 연구로 나눌 수 있다. 동시에 일정계획 수립의 구조(architecture)는 중앙 집중적 구조(centralized architecture), 분산 구조(distributed architecture), 계층적 구조(hierarchical architecture) 등이 있다. 이 두개의 차원에 의해 기존 연구를 분류하면, 다음 〈표 2〉와 같다.

일정계획 수립에 관한 최근 연구의 추세는 일정 계획 생성만 다루는 연구에서 일정계획 생성, 수정, 통제를 함께 다루는 연구로, 중앙 집중적 구조에서 분산 구조 또는 계층적 구조로 변화하고 있다. 본 연구는 일정계획 수정 및 통제에 관한 연구로서 계층적 구조를 갖는 쪽에 속하는데 아직까지 이 분야에 관한 연구가 비교적 미진한 편이다.

## 3. 조선 일정계획 수정을 위한 방법론

### 3.1. 조선 일정계획 수정의 목표와 제약 조건

일정계획 수정단계에 초기 탑재 일정계획을 수정하기 위해서, 우선 탑재 일정계획의 생성과 수정에 영향을 미칠 수 있는 탑재 일정계획의 특성을 파악해야 한다.

Architecture	Generative Scheduling	Generative Scheduling and Reactive Scheduling
Centralized	<ul style="list-style-type: none"> <li>· Fox and Smith, 1984 (ISIS)</li> <li>· Pape, 1985 (SOJA)</li> </ul>	<ul style="list-style-type: none"> <li>· Yamamoto and Nof, 1985</li> <li>· Lee and Suh, 1988 (PAMS)</li> <li>· Sarin and Salgame, 1990</li> <li>· Zweben et al, 1990, 1993</li> <li>· Li et al, 1993</li> <li>· Suh et al, 1993 (ROSE)</li> </ul>
Distributed (including Blackboard)		<ul style="list-style-type: none"> <li>· Smith et al, 1986</li> <li>· Smith, 1987</li> <li>· Ow et al, 1988</li> <li>· Smith et al, 1990 (OPIS)</li> <li>· Parunak et al, 1987 (YAMS)</li> <li>· Hynynen, 1988</li> <li>· Collinot et al, 1988, 1991 (SONIA)</li> <li>· Prosser, 1989</li> <li>· Bruke and Prosser, 1989, 1990 (DAS)</li> <li>· Dutta, 1990</li> <li>· Sycara et al, 1990, 1991</li> <li>· Anandhi et al., 1994</li> </ul>
Hierarchical		<ul style="list-style-type: none"> <li>· Suh, 1991</li> <li>· Lee et al, 1993, 1994</li> </ul>

〈표 2〉 일정계획 생성, 수정, 통제와 Architecture.

첫째, 탑재 작업은 순차적으로 이루어진다. 하나의 도크에는 한 개의 크레인만 존재하기 때문에 블럭과 슈퍼블럭은 한번에 하나씩 차례로 탑재될 수밖에 없다. 그러나 한 조선소에 두 개 이상의 도크가 존재하면 한 도크의 입장에서 보면 순차적으로 탑재가 이루어지나 조선소 전체적으로 보면 특정 시점에서 두 개 이상의 블럭이 탑재될 수도 있다.

둘째, 탐색 공간(search space)이 방대하다. 실제 현장에서의 탑재 일정계획 수립은 두 개 이상의 선박을 대상으로 하고 있으며, 또한 각각의 선박을 수백

개의 블럭으로 이루어져 있기 때문에 합리적인 탑재순서를 결정하기 위해서 고려해야 하는 탑재순서들의 조합은 상상할 수 없을 정도로 많다.

셋째, 탑재 순서를 결정할 때 고려해야 할 탑재 기술적 지식이 존재한다. 도크에서의 탑재 순서는 탑재기술자들의 머리 속에 있는 탑재 기술적 지식을 근거로 한 '초기 탑재블럭의 선정 기준'과 '블럭간의 탑재 선후관계'에 의해 결정된다. 초기 탑재블럭의 선정기준은 최초의 탑재 블럭은 엔진룸 혹은 중앙부의 바닥에 속하며 가장자리가 아닌 블럭 중에서

선정되어야 한다는 것이고, 블럭간 탑재 선후관계는 지지대를 필요로 하는 블럭은 지지대가 될 블럭이 먼저 탑재된 후에야 탑재가 가능하다는 것이다.

넷째, 조선소의 주요 자원은 인력이다. 조선에 소요되는 자원은 크게 인력, 크레인(예 : dock의 goliath crane), 작업공간(예 : PE shop의 정반) 등이다. 이 중에서도 인력이 중요한 자원으로 손꼽히는데 그 이유는 조선 생산공정의 대부분이 용접작업이며, 이러한 용접작업은 거의 사람에 의해 수행되기 때문이다. 따라서 균형 있는 인력의 활용은 전체적인 생산성 향상에 지대한 영향을 미치는 것으로 인식된다.

다섯째, 정반과 크레인의 제약은 인력제약보다 엄격하다. 인력의 경우 초과근무나 작업장간 인력이동을 통해 약간의 유동적 운영이 가능하지만, 크레인이나 작업공간이 없는 작업이 이루어 질 수 없기 때문에 특히 크레인과 정반은 효율적으로 관리되어야 한다.

여섯째, 일정계획 수정단계에서는 시간적 제약이 추가된다. 일정계획 생성단계에서 발생한 실패블럭을 제거하기 위해 일정계획 수정기는 하위 일정계획 생성기로부터 가능한 조립완료일을 넘겨받아 초기 탑재 일정계획을 수정한다. 또한 일정계획 생성단계에서 최소화된 전체공기(makespan)는 일정계획 수정단계에서는 고정된 것으로 보기 때문에 또 하나의

시간적 제약으로 작용한다.

이와 같은 특징으로부터 조선 일정계획 수정의 목표와 제약조건을 결정할 수 있다. 조선 일정계획 수정의 목표는

- ① 기간별 조업도를 평준화시키고
- ② 초기 탑재 일정계획의 변화를 최소화하면서
- ③ 실패블럭의 수를 최소화하는

탑재 일정계획을 부분적으로 수정하는데 있다. 그리고, 제약조건은 다음과 같다.

- ① 자원 제약조건 : 인력, 크레인, 정반
- ② 기술적 제약조건 : 초기 탑재블럭의 선정기준 블럭의 탑재 선후관계
- ③ 시간적 제약조건 : 실패블럭의 새로운 조립완료일, 주어진 전체공기

또한 일정계획 수정은 결국 일정계획 생성단계로부터 생성된 초기 탑재 일정계획을 부분적으로 수정하는 것이므로 각 단계의 목표와 제약조건은 밀접한 관계가 있다. <표 3>에서처럼 두 단계는 자원 제약조건과 기술적 제약조건을 공유한다. 반면에 일정계획 수정기는 탑재 일정계획 생성기에 의해 최소화되어 주어지는 전체공기와, 조립 일정계획 생성기에 의해 요구되는 가능한 조립완료일을 시간적 제약조건으로 추가해서 고려해야 한다.

	일정계획 생성	일정계획 수정
목 표	<ul style="list-style-type: none"> <li>① 기간별 조업도 평준화</li> <li>② 전체공기 최소화</li> </ul>	<ul style="list-style-type: none"> <li>① 기간별 조업도 평준화</li> <li>② 초기 탑재 일정계획의 변화 최소화</li> <li>③ 실패블럭의 수 최소화</li> </ul>
제약조건	<ul style="list-style-type: none"> <li>① 자원 제약조건 (인력, 크레인, 정반)</li> <li>② 기술적 제약조건 (초기 탑재블럭의 선정, 탑재 선후관계)</li> </ul>	<ul style="list-style-type: none"> <li>① 자원 제약조건 (인력, 크레인, 정반)</li> <li>② 기술적 제약조건 (초기 탑재블럭의 선정, 탑재 선후관계)</li> <li>③ 시간적 제약조건 (실패블럭의 조립완료일, 전체공기)</li> </ul>

<표 3> 일정계획 생성과 일정계획 수정의 목표 및 제약조건 비교

### 3.2. 조선 일정계획 수정의 알고리즘

조선의 탑재 일정계획 수정기는 하위 일정계획 생성기로부터 조립완료일을 만족시킬 수 없어 조정이 필요한 다수개의 실패블럭을 한꺼번에 넘겨받는다. 조정요청은 대개의 경우 블럭의 조립공장의 사정상 조립완료일을 일정기간 연기해 달라는 내용이다. 각 블럭의 조립완료일을 연기하면 각 공정의 작업시간을 감안해 볼 때 그 블럭의 탑재일 또한 연기되어야 하는데, 탑재되는 모든 블럭은 기술적 제약조건에 의해 탑재 선후관계로 묶여 있으므로, 한 블럭의 탑재일 연기는 여유시간(slack time)이 없는 한 후행 블럭 탑재일에 영향을 미치게 된다. 따라서 후행 블럭과의 여유시간을 발견하거나, 조립완료일과 탑재일 사이 공정의 작업시간을 단축하거나, 조립완료일과 탑재일 사이 공정의 버퍼(buffer, idle time)를 제거하는 과정 등이 필요하지만, 고려해야 할 블럭의 수가 워낙 많고, 다수개의 블럭의 각 공정이 서로 얽혀 있어 부분 수정이 매우 어렵다.

그러므로 다수개의 실패블럭을 한번에 모두 고려하는 것보다 우선 탑재일과 같은 적당한 기준에 따라 실패블럭을 순서대로 나열하고, 적용 가능한 일정계획 조정 방법을 수집하여 ‘수정 휴리스틱(Repair Heuristic)’을 만들고 이를 우선 순위에 따라 적용하는 것이 개념적으로 일정계획 수정과정상의 복잡성을 줄여줄 수 있었다.

이와 같은 착상을 바탕으로 하위 일정계획 생성기로부터의 조정요청을 반영하기 위해서 부분 수정을 통해 충격을 최소화시키는 일련의 휴리스틱으로 구성된 알고리즘을 제시하였다. 일정계획 수정은 크게 ‘현재블럭 선정 단계,’ ‘탑재일 조정단계,’ ‘각 공정 일정계획 조정단계’의 3부분으로 나누어지고, 각 step에 대한 자세한 설명은 다음과 같다. 또한 각각의 수정 휴리스틱에 대한 구체적인 내용은 다음 장에 기술했다.

#### 1) 현재블럭 선정 단계

- Step 1. 실패블럭들을 탑재일이 빠른 순서로 나열한다.
- Step 2. 실패블럭들 중에서 탑재일이 가장 빠른 블럭을 선택하여 현재블럭으로 삼는다.
- Step 3. 현재블럭이 일정계획 생성단계에서 사용했던 자원을 취소한다.
- Step 4. 현재블럭의 조정요구폭 계산한다.

#### 2) 탑재일 조정 단계

- Step 5. 현재블럭의 여유시간을 계산하고, 그 여유시간이 조정요구폭을 흡수할 수 있으면 step 10으로, 그렇지 않으면 다음 step으로 간다.
- Step 6. 현재블럭의 버퍼(buffer)를 찾아내고, 그 버퍼를 합한 누적조정폭이 조정요구폭을 흡수할 수 있으면 step 10으로, 그렇지 않으면 다음 step으로 간다.
- Step 7. 현재블럭의 작업시간을 일정비율 만큼 축소하고, 그 축소폭을 합한 누적조정폭이 조정요구폭을 흡수할 수 있으면 step 10으로, 그렇지 않으면 다음 step으로 간다.
- Step 8. 현재블럭의 다단계 여유시간을 계산하고, 그 여유시간을 합한 누적조정폭이 조정요구폭을 흡수할 수 있으면 step 10으로, 그렇지 않으면 다음 step으로 간다. 후행 블럭들 중에서 탑재일 조정이 필요한 블럭을 새로운 실패블럭으로 추가한다.
- Step 9. 현재블럭의 미 흡수 된 조정요구폭만큼 후행블럭의 탑재일이 조정 가능하면 현재블럭의 탑재일을 연기하고, 후행블럭들 중에서 탑재일 조정이 필요한 블럭은 새로운 실패블럭으로 추가한다. 만약 후행블럭의 탑재일을 조정할 수 없으면 현재블럭을 이중실패블럭으로 처리하고 step

14로 간다.

3) 각 공정 일정계획 조정 단계

- Step 10. 새로운 탑재일을 기준으로 현재블럭의 각 공정 일정계획을 수립한다. 만약 공정 일정계획 수립이 가능하면 step 14로, 그렇지 않으면 다음 step으로 간다.
- Step 11. 과부하가 걸리는 자원을 현재블럭과 공유하는 다른 블럭들 중, 여유시간 또는 버퍼가 있어서 일정계획 조정이 가능한 블럭을 찾아 조정한다. 만약 조정이 가능하면 step 14로 그렇지 않으면, 다음 step으로 간다.
- Step 12. 현재블럭에 대해 각 제약조건을 전략에 의해 미리 정한 수준의 제약조건 완화 방법을 적용한다. 만약 조정이 가능하면 다음 step 14로 그렇지 않으면, 다음 step으로 간다.
- Step 13. 현재블럭의 과부하가 해소될 때까지 후행블럭으로 전파가 가능하면 후행블럭으로 전파하고, 다음 블럭들 중에서 탑재일 조정이 필요한 블럭을 새로운 실패블럭으로 추가한다. 만약 전파할 수 없으면 현재블럭을 이중실패블럭으로 처리한다.

Step 14. 남아 있는 실패블럭이 없으면 stop하고, 그렇지 않으면 step 1로 간다.

3.3. 조선 일정계획 수정 휴리스틱

일정계획 수정 휴리스틱은 하위 조립 일정계획 생성기의 조정요구와 초기 탑재 일정계획의 현재 상태를 고려하여, 상황별로 적절한 수정 휴리스틱을 적용한다.

수정 휴리스틱은 단계별로 대략 ‘탑재일 조정’과 ‘각 공정 일정계획 조정’으로 구분된다. 즉, 먼저 탑재일 조정을 통해 탑재 선후관계를 고려하면서 조정요구폭을 만족하는 새로운 탑재일을 찾아내고, 그 탑재일을 기준으로 각 공정의 일정계획을 수립한 다음, 공정 일정계획 조정을 통해 자원 제약조건을 완화한다. 만약 조정이 불가능한 블럭이 발생하면 이중실패블럭(double failed block)으로 처리한다. 관련 내용은 다음 <표 4>에 요약되어 있다.

각각의 수정 휴리스틱에 대한 구체적인 설명은 다음과 같다.

1) 탑재일 조정

① 여유시간 제거

단 계	수정 휴리스틱	수정 범위
탑재일 조정	여유시간 제거	현재블럭과 후행블럭
	버퍼 제거	현재블럭
	공정별 작업시간 단축	현재블럭
	다단계 여유시간 제거	현재블럭과 후행블럭 모두
	후행블럭으로 전파	현재블럭과 후행블럭
각 공정 일정계획 조정	과부하 자원 공유블럭 조정	현재블럭과 과부하 자원 공유블럭
	자원 제약조건 완화	현재블럭
	후행블럭으로 전파	현재블럭과 후행블럭

<표 4> 수정 휴리스틱



여유시간 제거(simple slack time reduction)에서는 실패블럭의 직후 블럭들만을 고려하여 여유시간을 구하고, 그 여유시간 만큼 탑재일을 연기한다. 이 방법은 조정요구에 의해 발생한 충격을 현재 블럭과 직후 블럭만 고려하여 조정, 흡수하려고 하는 것이기 때문에 탑재일 조정방법 중에서 가장 간단하다. 대개의 경우 여유시간은 1~5일 정도의 값을 가진다.

## ② 버퍼 제거

버퍼 제거(buffer reduction or idle time reduction)에서는 실패블럭의 조립완료일 이후부터 탑재일 이전까지 기간 중에 버퍼가 있으면, 그 버퍼를 없앤다. 버퍼는 일정계획 생성단계의 제약조건 완화(constraint relaxation)에 의해 발생하며, 작업물량이 많을수록 커지고 대략 1~3일 정도의 값을 가진다. 이 역시 현재 실패블럭만 조정하는 간단한 방법이다.

## ③ 공정별 작업시간 단축

공정별 작업시간 단축(processing time reduction, or activity duration reduction)에서는 실패블럭의 조립완료일 이후부터 탑재일 이전까지 공정 중에서 작업시간을 줄일 수 있는 공정의 작업 시간을 일정비율만큼 단축시킨다.

단축폭은 조립완료일부터 탑재일까지의 기간이 길수록 커지므로 선행탑재 공정이 있는 슈퍼블럭이 일반블럭보다 큰 단축폭을 갖는다. 물론 각 공정별로 다르기는 하지만 작업시간 단축 비율은 표준 작업시간의 10~25% 정도이다. 이 역시 현재 실패블럭만 조정하는 간단한 방법이다.

## ④ 다단계 여유시간 제거

다단계 여유시간 제거(multi-slack time reduction)에서는 실패블럭 이후의 모든 블럭들을 고려하여

여유시간을 구하고, 그 여유시간 만큼 탑재일을 연기한다.

다단계 여유시간을 구하기 위해서는 가장 늦은 탑재 시작일(latest start time, LST)을 계산하여 실패블럭이 주경로(critical path) 상에 존재하지 않음을 확인해야 한다. 동시에 현재 실패블럭의 탑재일 조정에 의해, 후행블럭들 중에서 탑재일 조정이 필요한 블럭이 생기면 새로운 실패블럭으로 추가한다.

## ⑤ 후행블럭으로 전파

후행블럭으로 전파(propagation to succeeding blocks)에서는 실패블럭의 미 흡수된 조정요구폭 만큼 현재블럭의 탑재일을 일단 연기하고, 후행블럭들 중에서 탑재일 조정이 필요한 블럭이 생기면 새로운 실패블럭으로 추가한다. 만약 전체공기의 제약이나 새로운 실패블럭의 기하급수적 증가로 인해 전파가 불가능하면, 실패블럭을 이중실패블럭으로 처리한다.

## 2) 공정 일정계획 조정방법

### ① 과부하 자원 공유블럭 조정

과부하자원 공유블럭 조정에서는 과부하가 걸리는 자원을 실패블럭과 공유하는 다른 블럭들 중, 실패블럭과 탑재 선후관계가 없고, 여유시간 또는 버퍼가 있어서 일정계획 조정이 가능한 블럭을 찾아 일정계획을 조정한다.

### ② 제약조건 완화

제약조건 완화에서는 실패블럭에 대해 일정계획 생성단계에서 사용했던 제약조건 완화 방법을 적용하여 인력, 크레인, 정반의 제약을 전략에 의해 미리 정한 수준만큼 완화한다.

### ③ 후행블럭으로 전파

후행블럭으로 전파에서는 실패블럭의 과부하가 해소될 때까지 일정계획을 연기한다. 동시에 후행블럭들 중에서 일정계획 조정이 필요한 블럭들을 새로운 실패블럭으로 추가한다. 만약 전체공기의 제약이나 새로운 실패블럭의 기하급수적 증가로 인해 전파가 불가능하면, 실패블럭을 이중실패블럭으로 처리한다.

### 3.4. 조선 일정계획 수정 전략

탑재 일정계획 수정 시, 환경 또는 사용자의 의지에 의해 결정되어 일정계획 결과에 중대한 영향을 미칠 수 있는 변수를 가리켜 ‘조선 일정계획 수정 전략(Strategy for Reactive Scheduling)’라 한다. 수정 전략으로는 실패블럭의 최대 조정요구폭, 실패블럭의 수, 공정별 작업시간 단축의 최대 허용폭, 최대 조업도, 이중실패블럭의 처리 등이 고려될 수 있다.

#### 1) 실패블럭의 최대 조정요구폭

실패블럭의 최대 조정요구폭이 클수록 일정계획 수정기에서 그 조정요구폭을 반영하기가 어려워지므로 이중실패블럭이 많아진다. 하지만 최대 조정요구폭이 작으면 하위 일정계획 생성기에서 일정계획을 생성하는 동안 큰 조정요구폭을 해결하기 위해 여러 개의 작은 요구폭의 실패블럭이 많이 발생할 수 있다. 따라서 각 공장 별로 실패블럭의 발생에 의한 고통을 분담할 수 있도록 분산시키는 균형점(equilibrium point)을 찾을 수 있는데, 이것은 다양한 실험을 통해 가능하다.

#### 2) 실패블럭의 수

실패블럭의 수는 조립공장의 일정계획 수립 전략에 의해서 바뀔 수도 있지만, 작업물량과 같은 환경변

수에 영향을 받는 경우도 많다. 실패블럭이 많다는 것은 그만큼 일정계획 수정기에 부담을 줄 수밖에 없으므로, 실패블럭의 개수가 많을 때에는 자원 제약조건의 완화 등을 통해 일정계획 수정기의 해결의 폭을 넓혀 주어야 한다.

#### 3) 공정별 작업시간 단축의 최대 허용폭

각 공정별로 표준 작업시간의 10~25%로 되어 있는 공정별 작업시간 단축의 최대 허용폭을 변화시킴으로써 일정계획 결과에 큰 영향을 미칠 수 있다.

#### 4) 최대 조업도

현재 100~130%로 되어 있는 최대 조업도를 변화시킴으로써 일정계획 결과에 큰 영향을 미칠 수 있다.

#### 5) 이중실패블럭의 처리

이중실패블럭은 일정계획 수정기에서 일정계획 수정에 실패한 블럭을 말한다. 이러한 이중 실패블럭을 처리하기 위해서 첫 번째로 이 블럭을 하위 조립공장에 다시 보내는 방법이 있다. 즉 이 블럭에 대한 현재까지의 조정폭을 토대로 새로운 조립완료 요구일을 구하고 이것을 해당 조립공장에 넘겨주어 조립일정계획을 다시 수립하게 한다. 이러한 방법은 각 조립공장 자체의 일정계획 수정에 해당한다.

두 번째로 이중실패블럭을 일정계획 수정기가 포용하는 방법이 있다. 만약 이중실패블럭이 탑재일 조정단계에서 발생하지 않고, 공정 일정계획 조정단계에서 발생했다면, 자원 제약조건의 과부하를 비정상적인 초과근무 등으로 해결할 수 있다. 세 번째로는 블럭의 조립공장을 작업이 가능하면서도 여유가 있는 다른 조립공장으로 교체하거나 외주를 주는 방법이

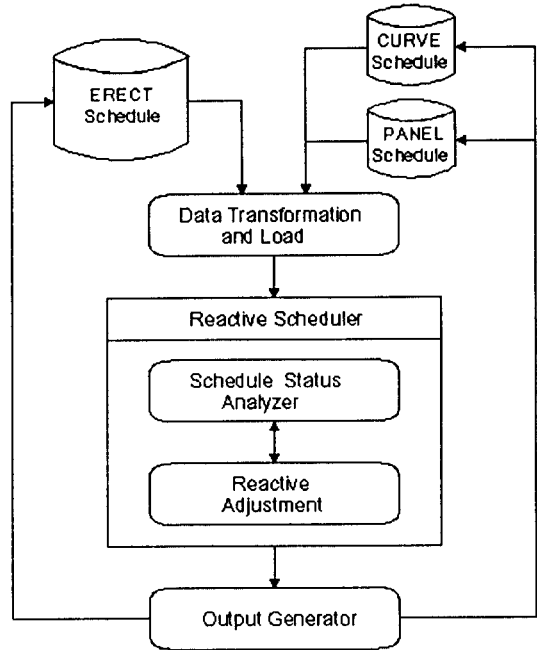
있다.

### 3.5. 조선 일정계획 통제

조선 일정계획 수정의 알고리즘은 계획단계(schedule-construction phase)의 일정계획 수정단계(reactive scheduling stage)의 일정계획 수정(reactive scheduling)에 뿐만 아니라, 비슷한 방법론을 통해 실행단계(schedule-operation phase)의 일정계획 통제(reactive control)에도 적용될 수 있다. 일정계획 통제는 주로 기계고장, 원자재 수급 지연, 파업, 태업, 기상이변 등에 의해 발생하며, 역시 현재의 탑재 일정계획을 수정해야 한다.

본 논문에서는 일정계획 수정에 중점을 두고 일정계획 통제에 대한 구체적인 방법론 설계와 구현은 향후 연구과제로 남겨 놓았다.

CURVE'를 역시 UNIK를 사용하여 개발하였다.



〈그림 3〉 DAS-REACT의 시스템 구조도

## 4. 일정계획 수정 시스템 DAS-REACT의 개발

### 4.1. 시스템 구조 및 지식 표현

위와 같은 조선 일정계획 수정 알고리즘을 대우 중공업 대우조선의 상황에 적용하여 일정계획 수정 시스템인 'DAS-REACT'를 개발하였다. DAS-REACT는 한국과학기술원 지능정보시스템 연구실과 대우중공업이 공동으로 진행중인 '대우조선의 일정관리 전문가시스템의 개발 : DAS 프로젝트'의 부분 시스템으로 전문가시스템 개발도구인 UNIK를 사용하여 SUN SPARC Station에서 구현되었다. DAS 프로젝트에서는 상위 탑재 일정계획 생성기인 'DAS-ERECT,' 평블럭 조립공장 일정계획 생성기인 'DAS-PANEL,' 곡블럭 조립공장 일정계획 생성기인 'DAS-

DAS-REACT의 시스템 구조도(system architecture)는 다음 〈그림 4〉과 같다. 또한 DAS-REACT에서 필요한 모든 정보는 상속과 디몬(demon) 등의 유용한 특성을 많이 갖고 있는 것으로 알려진 프레임(frame)의 형태로 〈그림 5〉처럼 표현하고, UNIK의 프레임 도구인 UNIK-FRAME을 이용하여 구현하였다.

한편 DAS-REACT는 DAS-ERECT, DAS-PANEL, DAS-CURVE의 세 가지 시스템 사이에서 발생할 수 있는 부조화를 해결해 주는 역할을 수행하므로, 세 시스템 모두와 밀접한 관계를 유지해야 한다. 즉, 상위 일정계획 시스템과 하위 일정계획 시스템들은 일정계획 생성 및 수정단계에 서로 필요한 정보를 교환한다. 이와 같은 시스템 사이의 메시지 교환(message transmission)을 지원하기 위해 〈그림 6〉처럼 프레임

형태의 메시지 프로토콜(message protocol)을 만들었다.

마지막으로 DAS-REACT가 역할을 수행하는 과정을 간략히 살펴보면, 먼저 DAS-ERECT의 초기

```

{{4032-62A
  is-a : SUPER-BLOCK
  part-of : PROJECT4032
  consist-of : 4032-621 4032-622
  shop : PBS

  previous-block : (4032-403 3) (4032-404 3) (4032-405 3)
  next-block : 4032-561 4032-562 4032-563 4032-623

  block-assembly : (DUR 19) (MH 995)
  block-outfitting : (DUR 3) (MH 13)
  block-painting : (DUR 8) (MH 460)
  PE-fwelding : (DUR 12) (MH 295)
  PE-inspection : (DUR 2) (MH 11)
  PE-repainting : (DUR 6) (MH 134)
  dock-fwelding : (DUR 13) (MH 770)
  block-assembly-sst : 93/11/18
  block-assembly-sft : 93/12/06
  block-outfitting-sst : 93/12/07
  block-outfitting-sft : 93/12/09
  block-painting-sst : 93/12/10
  block-painting-sft : 93/12/17
  PE-fwelding-sst : 93/12/18
  PE-fwelding-sft : 93/12/29
  PE-inspection-sst : 93/12/30
  PE-inspection-sft : 93/12/31
  PE-repainting-sst : 94/01/01
  PE-repainting-sft : 94/01/05
  erection-date : 94/01/06
  dock-fwelding-sst : 94/01/07
  dock-fwelding-sft : 94/01/19
  block-status : FAILED
  feasible-assembly-sft : 93/12/10
}}
* DUR : duration, MH : manhour
sst : scheduled start time, sft : scheduled finish time

```

<그림 4> 블럭정보의 프레임에 의한 표현

1) DAS-ERECT → DAS-PANEL, DAS-CURVE

```

{{4032-621
  is-a : BLOCK
  part-of : PROJECT4032
  shop : PBS

  block-assembly-sft : 93/12/08
}}
{{4032-622
  is-a : BLOCK
  part-of : PROJECT4032
  shop : 3DS

  block-assembly-sft : 93/12/06
}}

```

2) DAS-REACT ← DAS-PANEL, DAS-CURVE

```

{{4032-621
  is-a : BLOCK
  part-of : PROJECT4032
  shop : PBS

  block-status : SCHEDULED
  shop-out-date : 93/12/07
}}
{{4032-621
  is-a : BLOCK
  part-of : PROJECT4032
  shop : 3DS

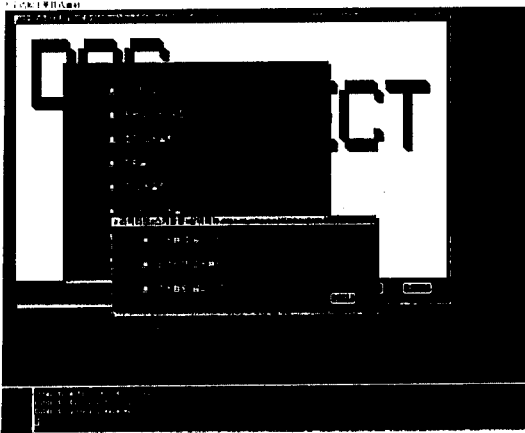
  block-status : FAILED
  shop-out-date : 93/12/10
}}

```

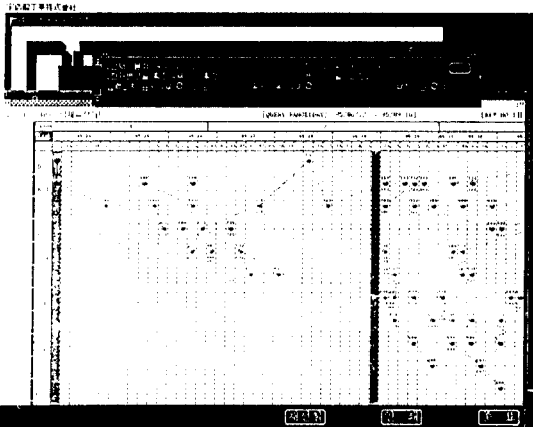
<그림 5> 메시지 프로토콜의 프레임에 의한 표현

탐재일정계획과 DAS-PANEL과 DAS-CURVE의 가능한 조립완료일을 메시지 프로토콜의 프레임 파일로 받아 DAS-REACT에서 사용할 프레임의 형태로 변환하는 작업을 한다. 변환된 프레임 파일을 적재(load)하고, DAS-ERECT의 결과로부터 관련정보를 추가로 적재한 후 본격적인 일정계획 수정에 착수한다. 일정계획 결과는 또다시 프레임 형태로 파일을 통해 출력된다.

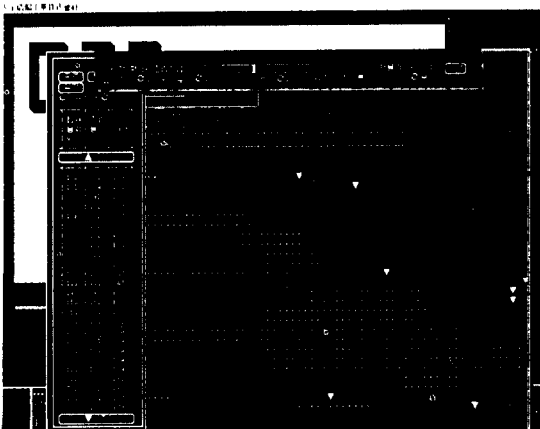
#### 4.2. 시스템 화면의 예



<그림 6> DAS-REACT의 메뉴 화면



<그림 7> 탑재일 조정 화면



<그림 8> 각 공정 일정계획 조정 화면

## 5. 결 론

본 연구에서는 조선의 계층적 일정계획 수립과정에서 발생할 수 있는 일정계획 수립기 사이의 일정계획 불일치를 제거하기 위해 일정계획 수정 알고리즘을 제시하였다.

이 알고리즘에 의하면 일정계획 수정기는 하위 일정계획 생성기로부터 조정요청을 받아 초기 탑재 일정계획을 부분적으로 수정하기 위해 충격을 최소화시키는 일련의 휴리스틱을 사용하였다. 각각의 휴리스틱은 여유시간 제거, 버퍼 제거, 작업시간 축소, 다단계 여유시간 제거, 과부하 자원 공유블럭 조정, 자원 제약조건 완화 등이고, 흡수되지 않은 일부의 충격은 주변으로 전파된다. 또한 이러한 수정 알고리즘은 계획단계의 일정계획 수정에 뿐만 아니라 실행단계의 일정계획 통제에도 적용될 수 있다.

위와 같은 방법론을 대우조선의 상황에 적용하여 조선 일정계획 수정 시스템인 DAS-REACT를 개발하였다. 실제 현장 자료를 이용한 실험을 통해 DAS-REACT는 상위 일정계획 생성기와 하위 일정계획 생성기 사이에 발생한 일정계획의 불일치를 조정하였다. 또한, 조정이 힘든 비정상적인 부하가 발생했을 때에는 이를 각 일정계획 수립기로 분산시키기 위한 기준을 제시함으로써 일정계획 수립기간의 연계를 지원할 수 있다.

### 참 고 문 헌

- 이재규 외, "대우조선의 일정관리 전문가시스템 개발 : DAS 프로젝트," 지능정보시스템, 1992, pp.9-32
- 이재규 외, 대우조선의 일정관리 전문가시스템(DAS)의 개발, 연구보고서, 한국과학기술원, 1994
- 이재규 외, 대우조선의 일정관리 전문가시스템(DAS II)

- 의 개발, 연구보고서, 한국과학기술원, 1995
- 최형림, 김호동, 양옥렬, 이정승, 이재규, "탑재 일정계획 수립을 위한 전문가 시스템의 개발: DAS-ERECT," 한국전문가시스템학회 학술대회 논문집, 1993, pp.127-137
- Anandhi, B., S. V. Ajay, and S. Arun, "Blackboard Architecture for Reactive Scheduling," *Expert Systems with Applications*, 1994, pp.55-65
- Bruke, P., and P. Prosser, "A Distributed Asynchronous System for Predictive and Reactive Scheduling," Technical Report AISL-42-89, Department of Computer Science, University of Strathclyde, Glasgow, October 1989
- Bruke, P., and P. Prosser, "Distributed Asynchronous Scheduling," *Application of Artificial Intelligence in Engineering V, Vol.2 Manufacturing and Planning*, Computational Mechanics Publications, Southampton Boston, 1990, pp. 503-522
- Choi, H. R., *Erection Scheduling at Shipbuilding Using Constraint Directed Graph Search: DAS-ERECT*, Ph. D. Thesis, KAIST, 1993
- Collinot, A., and C. L. Pape, "Adapting the behavior of a Job-Shop Scheduling System," *Decision Support Systems*, Vol.7, 1991, pp.341-353
- Dutta, A., "Reacting to Scheduling Exceptions in FMS Environments," *IIE Transactions*, Vol.22, No.4, 1990, pp.300-314
- Fox, M. S., and S. F. Smith, "ISIS-A Knowledge-Based System for Factory Scheduling," *Expert Systems*, Vol. 1, No.1, 1984, pp.25-49
- Lee, J. K., and M. S. Suh, "PAMS: A Domain-Specific Knowledge-Based Parallel Machine Scheduling System," *Expert Systems*, Vol.5, No.3, August 1988, pp. 198-214
- Lee, J. K., M. S. Suh, and M. S. Fox, "Contingencies for the Design of Scheduling Expert Systems," *Expert Systems with Applications*, Vol.6, No.3, 1993, pp.219-230
- Lee, J. S., *Reactive Scheduling and Control for Shipbuilding*, Master's Thesis, KAIST, 1994
- Li, R., Y. Shyu, and S. Adiga, "A Heuristic Rescheduling Algorithm for Computer-based Production Scheduling Systems," *International Journal of Production Research*, Vol.31, No.8, 1993, pp.1815-1826
- Ow, P. S., S. F. Smith, and A. Thiriez, "Reactive Plan Revision," *AAAI*, 1988, pp.77-82
- Pape, L. C., "SOJA: A Daily Workshop Scheduling System," *Proceedings of the Fifth Technical Conference of the BCS Specialist Group on Expert Systems*, Warwick, Great Britain, 1985
- Prosser, P., "A Reactive Scheduling Agent," *Proceedings of International Joint Conference on Artificial Intelligence*, 1989, pp.1004-1009
- Sarin, S. C., and R. R. Salgame, "Development of a Knowledge-based System for Dynamic Scheduling," *International Journal of Production Research*, Vol.28, No.8, 1990, pp.1499-1512
- Smith, S. F., "A Constraint-Based Framework for Reactive Management of Factory Schedules," *Proceedings International Conference on Expert Systems and the*

Leading Edge in Production Planning and Control, May 1987, pp.1-17

Smith, S. F., M. S. Fox, and P. S. Ow, "Constructing and Maintaining Detailed Production Plans : Investigations into the Development of Knowledge-Based Factory Scheduling Systems," AI Magazine, Fall 1986, pp.45-61

Smith, S. F., P. S. Ow, J Potvin, N. Muscettola, and D. C. Matthys, "An Integrated Framework for Generating and Revising Factory Schedules." Journal of Operational Research Society, Vol. 41, No. 6, 1990, pp.539-552

Suh, M. S., A Hierarchical Job Shop Scheduling Expert System : KAIS-3, Ph. D. Thesis, KAIST, 1991

Suh, M. S., C. H. Kim, Y. K. Ko, and K. S. Lee, "ROSE : An Expert System for Reactive Adjustment of Hot-Rolling Schedules," CPC-93, 1993, pp.271-281

Sycara, K., S. Roth, N. Sadeh, and M. Fox, "Distributed Constrained Heuristic Search," IEEE Trans. on Systems, Men, Cybernetics, Vol. 21, No. 6, 1991, pp.1446-1461

Sycara, K. P., S. F. Roth, N. Sadeh, and M. S. Fox, "Distributed Production Control," Proceedings of the

Fourth International conference on Expert Systems in Production and Operations Management, 1990, pp. 80-89

Sycara, K. P., S. F. Roth, N. Sadeh, and M. S. Fox, "Resource Allocation in Distributed Factory Scheduling," IEEE Expert, February 1991, pp.29-40

Yamamoto, M., and S. Y. Nof, "Scheduling/Rescheduling in the Manufacturing Operating System Environment," International Journal of Production Research, Vol. 23, No. 4, 1985, pp.705-722

Zweben, M., E. Davis, B. Daun, and M. J. Deale, "Informedness vs. Computational Cost of Heuristics in Iterative Repair Scheduling," Proceedings of International Joint Conference on Artificial Intelligence, 1993, pp. 1416-1422

Zweben, M., E. Davis, B. Daun, and M. J. Deale, "Scheduling and Rescheduling with Iterative Repair," IEEE Trans. on Systems, Men, Cybernetics, 23(6), 1993, pp. 1588-1596

Zweben, M., M. Deale, and R. Gargan, "Anytime Rescheduling," Proceedings DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control, Morgan Kaufmann, San Mateo, CA, 1990