

☒ 연구논문

IMS(지능생산시스템)의 효율적인 운용 기법 개발에 관한  
연구 - 자원 관리 기법을 중심으로 - \*\*  
- A Method for Efficient Operation of Intelligent  
Manufacturing Systems -

이 영 해\*  
Lee, Young Hae  
이 병 호#  
Lee, Byung Ho  
조 창 희\*  
Cho, Chang Hee

Abstract

Manufacturing systems are becoming intelligent, decentralized, autonomous, which is so called as Intelligent Manufacturing Systems(IMS). In this paper, a method for efficient resource management in Holonic Manufacturing System which is one alternative of IMS, was developed. The developed method was evaluated with the current possible method and the results are shown.

1. 서 론

소비자들의 기호의 변화에 따른 급작스런 제품 모델 변경에의 대처와 신속한 생산을 통한 납기 만족은 오늘날의 생산시스템에서 매우 중요한 관심이 되고 있다. 또한, 컴퓨터 및 통신 기술의 발달에 힘입어 생산시스템의 기능이 매우 진보되었고 시스템의 내외 환경 변화에 대해 민첩하게 적응하고, 지적, 자율화 되었으며, 정보들이 분산되어 관리되는 등 매우 고도로 발달된 시스템으로 진보하기에 이르렀다. 따라서, 기능적인 면에서 매우 진보되고 지적이고, 자율화 되고 분산된 생산시스템을 적절히 통제하여 시스템이 제 기능을 발휘할 수 있도록 도와 주는 적절한 시스템 제어 및 운영 기술이 필요하게 되었다[1].

FMS, CIM에 이은 차세대 생산시스템으로 국내외적으로 IMS(Intelligent Manufacturing Systems)가 거론 되고 있으며, 이를 구현시키기 위한 활동의 일환으로 국제적으로는 IMS 국제공동프로젝트가 진행되고 있으며, 국내에서는 국가 주도의 G7 프로젝트에 1997년부터 시범적으로 수행되고 있다[1].

기존의 생산시스템의 제어 형태는 크게 계층 구조(hierarchical architecture)와 수평 구조(heterarchical architecture)로 구분된다. 그런데, 계층 구조는 안정성이 있으며 시스템의 최적해를 줄 수 있는 반면 동적 환경에서의 적용이 매우 어렵고, 수평 구조는 동적 환경 변화에 적용이 가능하지만 시스템의 최적화를 실현하기가 어려운 것으로 알려져 있다. 생산시스템의 기

\*\* 본 연구는 1995년도 한국학술진흥재단 대학부설연구소과제 연구비에 의하여 연구되었음

\* 한양대학교 산업공학과, # 한양대학교 전자공학과

존의 관리 구조는 미래의 생산환경 즉, 빠르게 변화하는 생산 환경에 대한 신속한 대처 능력과 시스템 목표의 달성에 대한 요구를 적절히 반영하지 못한다. 이런 문제점을 해결하기 위해 차세대 생산시스템으로 부상한 홀로닉 생산시스템에 대한 연구가 미국, 유럽 그리고 일본 등의 선진국에서 시작되었으나 아직 체계를 정립하는 초기 단계에 머무르고 있다. 홀로닉 생산시스템의 효율적인 자원 관리 운영시스템의 개발은 매우 중요하다[1].

홀론(Holon) 개념을 이용한 홀로닉 생산시스템(Holon Manufacturing System: HMS)[2,3,4]은 계층 구조의 빠른 반응시간과 수평 구조의 환경 변화에 민첩하게 적응하는 유연성 및 자율성의 특징을 나타내는 제어 구조를 갖도록 설계된다. 이를 위하여 시스템의 각 요소들에 대하여 모듈화가 이루어지고 정보들이 분산된다. 또한 각 모듈들의 정보들이 모듈 관리자들에 의해 수정되고 의사 결정 문제의 해결을 위해서는 서로 정보가 교환된다. 이처럼 홀로닉 생산시스템은 자율화, 협력화 방식으로 운영되어 환경 변화에 대해서 빠른 반응시간을 보장하고 유용한 자원을 충분히 이용함으로써 환경 변화에 적응하면서도 전체 시스템의 운영 최적화를 가능하게 한다[1].

본 연구에서는 여러 자원들로 구성된 생산시스템이 환경의 동적인 변화에 따라 신속하고 유연하게 대처하는 운영문제를 자율적으로 통제하는 홀로닉 생산시스템의 자원 할당 운영 기법에 관한 문제를 다루었다. 이를 위하여 생산시스템의 모델링 및 생산자원의 복잡도를 고려한 Resource 홀론간의 재협상을 통해 자원의 재배분을 가능하도록 하는 새로운 협상구조와 알고리즘을 개발하였다.

기존 연구로서, Tonshoff 등[8]은 홀로닉 생산시스템을 위한 제품모델링 절차를 제시하였으며, Agre 등[4]은 철강제품의 냉각온도 조절을 위해 홀로닉 생산시스템의 협상(negotiation) 알고리즘을 제시하였다. Valckenaers 등[9]은 홀론 구조의 효율성을 평가하기 위해 기존 관리 구조와 비교한 결과를 제시하였다. Wyns 등[11]은 홀로닉 생산시스템에서의 자원할당에 관한 홀론의 구성과 운용방법을 제시하였다. 그러나, 현재까지 명확한 결과를 제시한 연구는 없으며 단지 홀로닉 생산시스템의 타당성을 검토하기 위한 시험 단계에 머무르고 있는 실정이다.

본 논문에서는 분산-협조형 제어 구조의 특성을 가지는 홀로닉 생산시스템의 중요한 특성인 홀론간의 협상 방법을 모델링하기 위해 contract net 모형의 개념을 이용하였는데, 이와 관련된 연구는 다음과 같다. Davis와 Smith[5]는 분산시스템의 문제 해결을 위한 새로운 방법으로서 contract net 이라는 모형을 제시하였다. Contract net 은 노드와 메시지로 이루어지며 각 노드는 메시지교환을 통하여 서로 통신을 한다. Contract net 에서는 각 노드의 통신시에 각 노드 사이에서 일어나는 각종 메시지와 제어지시의 전달 등을 협상이라는 방법을 통하여 모형화하였으며 그림 1.1과 같다

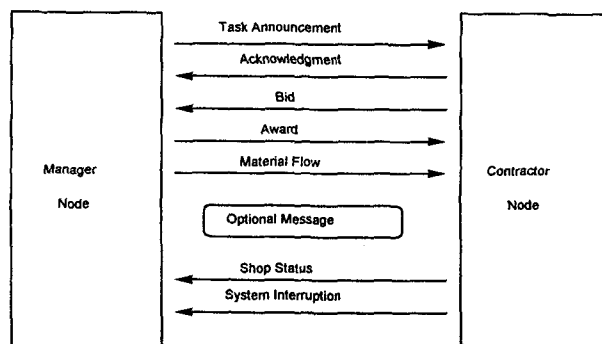


그림 1.1 Contract net 모형

분산 시스템의 특성상, 시스템 차원의 정확한 해를 구하기 위해서는 각 노드사이의 협조 (cooperation)에 의한 문제해결이 필요하다. 이때 각 노드가 어떻게 무슨 내용을 주고 받는지를 규정해 주어야 한다. 이런 방식들을 체계적으로 규정해 놓은 것이 바로 협상의 개념이다. 협상의 개념을 이용하여 입찰네트워크상에서 업무를 처리하고자 하는 경우 처리해야 될 업무를 정의하는 Manager 노드와 bid를 내는 contractor 노드가 정의된다.

Shaw[7]는 cell형 생산시스템에 LAN이 설치되어있는 경우를 대상으로 하여 분산형 동적 (distributed-dynamic) 생산일정계획에 관한 연구결과를 발표하였다. 이 연구에서 Shaw는 Davis와 Smith가 제시한 contract net 모형을 이용하여 네트워크 차원의 입찰방법을 사용하여 컴퓨터 통신망의 특성을 활용할 수 있는 분산 생산일정계획을 시도하였다. 또한 Shaw는 simulation을 통하여 cell제어용 컴퓨터에서 생산일정계획을 일괄적으로 수행하던 기존의 방법과 분산 생산일정계획의 방법에 대하여 비교분석을 하였다. 그 결과 분산 생산일정계획이 기존의 생산계획 방법보다 우월함을 보였고, 분산생산일정계획중에서도 EFT(earliest finish time)에 의한 자원 할당 방법이 좋은 결과를 보이고 있음을 밝혔다.

문[2]은 할당 문제(assignment problem)를 이용하여 부품과 자원의 할당에 대한 입찰문제를 풀었다. 문장식의 모형은 실행의 관점에서 볼 때 분산형 모형이라고 하기보다는 오히려 정보의 집중화에 의한 중앙 집중형 구조를 취하고 있다. Park[6]은 Davis와 Smith의 입찰 네트워크 모형[5]을 확장하여 단순한 modeling construct를 의미하는 노드의 개념 대신에 특정목적의 기능이 추가된 보다 발전된 개념의 모형화 도구인 agent의 개념을 사용하여 order agent (OA), resource agent(RA), communication forwarding agent(CFA)의 세 종류의 agent로 구성된 Multiple-agent를 가진 분산-협조형 제어 시스템을 제안하였다. 또한 자원 할당을 위해 resource congestion level을 이용한 협상 알고리즘을 개발하였다. 그러나 Davis와 Smith가 제안한 contract net을 사용한 대부분의 연구의 자원 할당은 자원간의 재협상을 간과하고 있다.

## 2. 홀로닉 생산시스템

### 2.1 홀로닉 생산시스템의 구성요소

홀로닉 생산시스템은 시스템 구성 요소들이 자율적이면서도 협조적인 특징을 가지고 있는 구조로서 아래와 같은 내용들을 포함한다.

- 1) 홀로닉 생산시스템은 홀론들로 구성된다. 홀론은 제조 시스템에 있어서 자율적 이면서도 협력적인 성질을 가진 시스템의 구성요소 및 과정으로서 제품의 전환, 운송및 저장, 정보의 가공 처리, 생산 계획, 생산 일정 계획, 생산 통제일 수도 있다. 또한 홀론은 정보 처리 부분과 물류 처리 부분으로 나누어질 수 있으며, 하나의 홀론은 다른 홀론의 부분으로도 될 수 있다.
- 2) 홀로닉 생산시스템을 구성하는 홀론들은 자율성(Autonomy)을 갖는다. 자기의 계획, 전략 및 운영을 스스로 수행할 수 있는 능력을 갖는다.
- 3) 홀론들은 시스템 공동의 목표 수행을 위해 서로 협조(Cooperative)하여 필요한 업무들을 수행한다.
- 4) Hierarchy 구조를 갖는다. 홀라키 구조는 공동의 목표 수행을 위해 협력 할 수 있는 홀론들의 구조로서 Holon 들의 협력을 위한 기본적인 규칙(basic rules)을 정의한다. 따라서 홀라키 구조는 각 홀론의 자율성(Autonomy)을 어느 정도 제한한다.

현재까지 홀로닉 생산시스템에 대한 연구가 세부분야별로 계속 연구되어지고 있으나 정확한 결과를 제시한 연구는 없으며 단지 홀로닉 생산시스템의 타당성을 검토하기 위한 시험 단계에 머무르고 있는 실정이다. 그런데 현재까지 발표된 연구 내용을 통해서 보건대 홀로닉

생산시스템의 구성을 위해 갖추어야 할 요소를 다음과 같이 세 가지로 요약할 수 있다. 첫째, 시스템의 목표에 영향을 줄 수 있는 시스템 요소들에 대한 홀론을 구성하여야 한다. 둘째, 시스템의 목표 성취를 위해 각 홀론들의 역할이 명확하게 정의되어야 한다. 각 홀론은 어떤 정보를 유지해야 하고 또 누구에게 자신이 가지는 정보를 어떤 형태로 제공해야 하는지, 다른 홀론들과 어떤 관계를 갖게 되는지 등이 명확하게 정의되어야 한다. 셋째, 홀론 간의 협력 및 협상을 위한 룰이 명확하게 정의되어야 한다.

## 2.2 홀론의 정의 및 구성

홀로닉 생산시스템은 의사 결정을 요구하는 일이 있을 때 여러 이해관계가 있는 홀론들이 유기적으로 결합되어 의사결정을 할 수 있는 홀론들로 구성된다. 홀론들은 서로 의사 정보를 교환함으로써 자신에게 부과된 업무를 수행해 가는데 시스템의 목표 수행을 위해 적절한 홀론을 구성하고 홀론 간의 정보교환이 원활하게 이루어지도록 생산시스템이 모델링되는 것이 필요하다. 홀로닉 생산시스템을 위한 홀론의 기본적인 구성은 product 홀론, order 홀론, resource 홀론과 의사결정을 지원하는 staff holon으로 구성되며 그림 2.1과 같다.

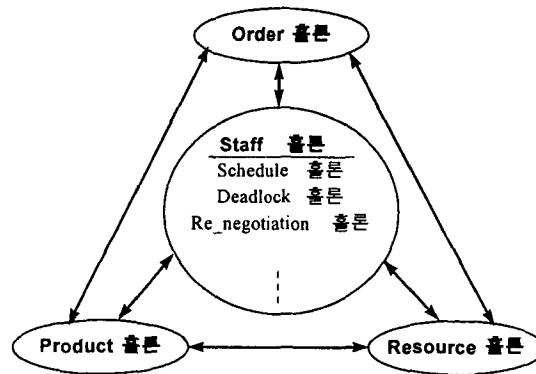


그림 2.1 홀로닉 생산시스템의 기본적인 홀론의 구성

홀로닉 생산시스템에서 자원 관리 문제는 기존의 생산시스템보다는 더욱 유연하고 동적이며 시의적절하게 생산에 필요한 자원을 운용, 통제하고 작업에 할당할 수 있도록 하는 자원 관리 시스템의 모델의 설계를 요구한다. 따라서 홀로닉 자원 관리 시스템의 모델을 설계하기 위해서 모델을 구성하는 홀론들을 정의한다. 홀론구조에는 기본 홀론과 Staff 홀론으로 구성되는데 기본 홀론에는 다음의 3가지로 구성된다.

- 1) Product 홀론 - 제품에 관한 정보를 관리하는 홀론.
- 2) Order 홀론 - 주문에 관한 정보를 관리하는 홀론
- 3) Resource 홀론 - 자원에 관한 정보를 관리하는 홀론.

또한 자원관리 시스템에서 의사결정을 지원하는 Staff 홀론은 다음의 3가지로 구성하였다.

- 1) Schedule 홀론 - 작업의 대한 생산 일정을 관리하는 홀론.
- 2) Deadlock Prevention 홀론 - deadlock 상태를 방지하는 홀론.
- 3) Re\_negotiation 홀론 - resource 홀론간의 재협상을 중재하는 홀론.

각 홀론의 역할과 입출력 정보를 세부적으로 설명하면 다음과 같다.

- (1) Product 홀론

고객의 주문처리를 위한 제품에 관한 모든 정보를 관리 및 제공한다. 제품의 종류, 특성, 제품을 구성하는 부품, 필요한 작업, 고객 요구사항, 제품 설계 사양, 공정 계획과 재료 명세서들에 대한 정보를 지속적으로 갱신 및 관리한다.

- 입력 정보 : 제품명, - 출력 정보 : 제품을 구성하는 부품, 필요한 작업, 작업 시간

(2) Order 홀론

수행해야할 주문들에 대한 정보와 각 홀론들로부터 주문의 처리 정보들을 갖고 각 주문들에 대하여 납기를 만족시킬 수 있는 의사결정에 관여한다. Order 홀론은 Product 홀론으로부터 주문된 제품에 대한 작업들에 관한 정보를 획득하여 Resource 홀론에 알린다. 이는 모든 자원을 대상으로 하는 것이 아니라 작업을 처리할 수 있는 Resource 홀론에게만 message를 보내고 만족할 만한 입찰조건을 낸 자원을 할당한다.

- 입력 정보: 주문 납기, 주문별 제품명, 제품별 작업 종류, 각 작업을 처리할 수 있는 자원  
- 출력 정보: 작업의 납기, 작업 시간,

(3) Resource 홀론

자원에 대한 physical part와 자원을 운용하는 information processing part로 구성되어 있다. 전자는 resource에 대한 생산 용량과 기능들을 제공한다. 후자는 작업을 수행하기 위해 resource들을 구성하고, 사용하고, 제어하는 knowledge와 procedure를 갖고 있다. Order 홀론으로부터 제안된 작업을 목적함수에 의하여 평가하고 입찰 조건을 낸다.

- 입력 정보: 작업의 납기, 작업 시간, 작업 할당 정보  
- 출력 정보: 자원의 상태, 자원의 용량, 작업의 처리가능 시간

(4) Schedule 홀론

작업에 대한 자원의 할당에 관한 전반적인 생산일정을 관리하며 시스템 이상 상태로 인한 일정 변경시 이를 각 홀론에 알린다.

- 입력 정보: 현장 상태, 자원의 상태, 작업 개시 및 완료, 작업에 할당된 자원  
- 출력 정보: 이상상태에 의한 변경된 일정

(5) Deadlock prevention 홀론

Deadlock 상태가 발생하지 않도록 schedule홀론으로부터 할당된 자원의 정보를 획득하여 deadlock 상태 여부를 예측하고 방지한다. 또한 이미 발생된 deadlock을 처리하는 역할.

- 입력 정보: 자원이 할당된 단계의 상태 정보, deadlock 상태,  
- 출력 정보: 변경된 자원할당

(6) Re\_negotiation 홀론

목적 함수를 만족시킬 수 있는 자원을 할당하지 못했을 경우, 모든 자원을 대상으로 이미 할당된 작업중 여유 있는 작업들을 조정하여 자원들간의 재협상을 시도하여 만족된 자원의 정보를 schedule홀론에 전달하는 역할을 한다.

- 입력 정보: 자원 상태, 각 자원의 작업 처리 가능 시간, 작업의 납기  
- 출력 정보: 작업에 할당된 자원, 기존 할당중 변경된 작업

(7) Repair 홀론

기계고장이 발생하였을 경우 예상 수리시간을 산출하고 기계고장의 수리를 담당하는 홀론이다.

- 입력 정보: 자원 상태, 고장 시간, 고장의 종류,  
- 출력 정보: 고장에 따른 수리 시간, 기계의 수리 방법

### 3. 자원 할당을 위한 알고리즘

본 절에서는 홀로닉 생산시스템하에서 입찰 모형을 이용한 자원 할당 절차를 정의하고 이를 지원하는 협상 및 재협상 알고리즘을 제안한다. Order 홀론에서는 수행해야할 작업들에

대한 정보를 각 resource 홀론에 알려주는 task announcement vector (TAV)를 선언한다. 동시에 resource 홀론에서는 event가 생길 때마다 각 자원의 상태를 기록한 shop status vector (SSV)를 schedule 홀론에 보낸다. Order 홀론이 task announcement를 할 때는 반드시 사전에 각 자원의 상태를 참고하여 작업 가능한 resource 홀론에 전달한다. Resource 홀론은 각자의 목적함수에 따라 작업에 대한 작업완료시간을 계산하여 order 홀론에 bidding vector (BV)를 보낸다. Order 홀론은 목적함수에 따라 BV중 적합한 것을 선택한다. 선택된 resource 홀론은 자신이 할당되었음을 schedule 홀론에 SSV를 보내고 작업을 마치면 다시 SSV를 보낸다.

본 논문에서는 작업할당 전략으로 생산량의 최대화(Max throughput)와 제품의 납기를 준수하여 생산하는(Min number of tardy job)전략을 채택한다. 또한 각 resource 홀론의 bid를 결정하는 목적함수가 비용, 수명, 운영 방침 등의 자원의 특성에 따라 다른 목적함수를 가질 수 있으나 모든 resource 홀론이 동일한 목적함수를 가지고 있는 것으로 가정하였다. 각 message에 대한 vector는 다음과 같이 정의한다.

- TAV : message id #, part #, operation #, (Routing (machine, processing time), ... (machine, processing time)), start time, finish time, least finish time
- BV : message id #, message id # for TAV, resource #, earliest start time, expected finish time
- SSV : resource #, resource status(가동, 대기, 고장), loading part #, loading time, expected release time, # of job in queue

홀론 구조에서의 제안된 자원 할당 절차를 그림으로 표현하면 그림 3.1과 같이 된다. resource 홀론복수의 bid들은 order 홀론과 resource 홀론간의 협상에 의해 조정되며, 각 resource 홀론들간의 재협상을 할 수 있는 재협상 알고리즘을 제시한다. Resource 홀론사이의 재협상 여부를 판정하기 위한 기준으로 자원의 복잡도(resource congestion level)를 정의하여 사용하였다. 어떤 작업 A가 자원에 할당되었을 때, 자원의 휴지시간이 전혀 없다는 전제하에

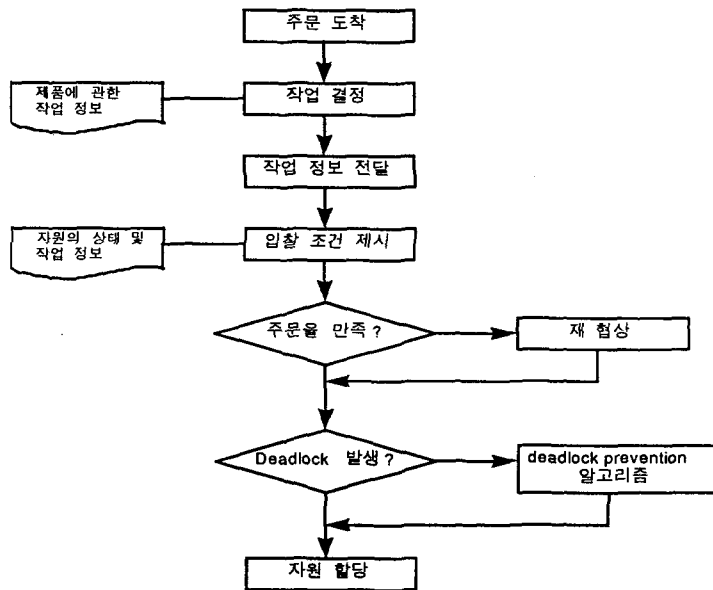


그림 3.1 자원 할당의 절차 흐름도

그 자원에 할당된 작업을 모두 처리한다고 할지라도 작업 A를 납기안에 마칠 수가 없는경우에 그 기계는 복잡한 상태에 있다고 정의한다.

본 논문에서 사용하는 용어의 정의는 다음과 같다.

$$OQT_i = \sum_{j=1} t_j$$

여기에서,  $j$ 는 자원  $i$ 의 queue 에 대기하는 작업,  $t_j$ 는 작업  $j$ 에 필요한 작업시간,  $OQT_i$ 는 자원  $i$ 에 할당된 작업들이 연속하여 가공이 이루어질 때의 가공시간의 합을 의미한다.

$$ERT_i = OQT_i + t_k$$

여기에서,  $k$ 는 기계  $i$ 에서 현재 가공중인 작업,  $ERT_i$ 는 자원  $i$ 가 가장 빨리 free 상태로 될 수 있는 시간을 의미한다.

$$EFT_i = ERT_i + t_{ann} (1 + a\%)$$

여기에서,  $t_{ann}$ 는 order 홀론에서 가공을 의뢰한 작업의 가공시간을 의미한다. 이 때  $EFT_i$ 는 TAV에 의해 의뢰된 작업이 자원  $i$ 에서 가장 빨리 가공이 완료될 수 있는 시간 (여유시간 ( $a\%$ ) 포함)을 의미하며 이 값이 납기보다 크면 자원  $i$ 가 복잡한 상태에 있다고 정의한다. 만약 어느 자원이 복잡한 상태에 있다면 그 자원은 작업을 가급적 할당하지 않고 대신 대체 공정상의 다른 자원을 이용하도록 한다. 그러나 모든 자원이 복잡한 상태에 있을 수 있으며 이런 경우  $a$ 값을 줄여가며 가능한 기계를 찾아본다. 어떤 자원도 주어진 기간내에 생산을 완료할 수 없을 경우 재협상 과정에서 각 resource 홀론에 할당된 작업중 가장 마지막에 할당된 작업을 제거한다.

이와 같은 개념을 이용하여 resource 홀론과 order 홀론간의 협상 및 resource 홀론끼리의 재협상을 지원하는 알고리즘을 제시하면 다음과 같다.

- 단계 1. 주어진 operation ( $t_{ann}$ )을 처리할 수 있는 대체 공정상의 모든 resource 홀론, 즉 TAV를 받은 모든 resource 홀론에 대하여 EFT 납기를 만족하는 resource 홀론이 존재하는 경우 그 resource 홀론을 선택하고 재협상을 마친다. 그렇지 않으면 단계 2로 간다.
- 단계 2. 단계 1의 모든 resource 홀론에 대하여  $W_i = ERT_i - t_j$ 를 계산한다.  
단  $t_j$ 는 resource 홀론에 가장 최근에 할당된 작업의 가공시간을 의미한다.
- 단계 3.  $W_i$ 와  $t_j$ 를 오름차순으로 정렬하여  $W_i$ 의 최소값에 해당하는 resource 홀론을  $X$ ,  $t_j$ 의 최소값에 해당하는 작업을  $Y$ 로 둔다.
- 단계 4.  $Y$ 를  $X$ 에서 가공할 수 있다면 단계 7을 수행하고, 그렇지 않으면 단계 5로 간다.
- 단계 5. 단계 3의 모든  $t_j$ 에 대하여  $Y' = \text{operation}[\min\{t_j\}]$ 를 구한다. (단  $Y \neq j$ )
- 단계 6.  $Y'$ 가 공집합이 아니면 단계 4를 수행하고, 공집합이면 단계 10으로 간다.
- 단계 7.  $Z = W_i(X) + t_Y + t_{mh} + t_{ann}$ 를 계산한다 ( $t_{mh}$ 는 자원사이의 이동시간).  
계산한  $Z$ 값이 납기보다 크면, 단계 10으로 가고 그렇지 않으면 단계 8로 간다.
- 단계 8. 두 개의 작업사이의 할당관계를 바꾼뒤에  $X_j$ 와  $Y$ 의 복잡도(criticality)를 계산한다.  
 $X_j$ 는 resource 홀론  $X$ 에 배정되었다가 제거된 작업을 의미한다. 계산 결과 두 작업중 하나라도 복잡한 상태라면 단계 9로 가고 그렇지 않다면 두 작업의 교환을 인정하고 재협상을 마친다.
- 단계 9.  $X_j$ 와  $Y$ 의 복잡도의 합이  $\{\min[EFT_i] - \text{납기}\}$ 보다 크면 단계 10으로 그렇지 않으면 두 작업의 교환을 인정하고 재협상을 마친다.
- 단계 10.  $\text{Min}[EFT_i]$ 을 만족하는 resource 홀론을 선택하고 재협상을 마친다.

본 논문에서 제시된 홀론식 생산시스템하에서의 분산-협조형 제어구조에서는 resource 홀론끼리 재협상을 통하여 이미 할당된 작업을 변경할 수 있는 기능을 가지고 있기 때문에 동적인 변화에 대해서도 효과적으로 대처할 수 있다. 그렇지만 자원과 작업사이의 할당관계를 바꾸어서 복잡도를 해결하였다고 하더라도 그에 따르는 역작용이 생길 수 있다. 즉 자원  $i$ 에서 제거된 작업이 자원  $j$ 에 할당되었을 때 오히려 복잡도가 커질수도 있다. 뿐만 아니라 기술적 제약도 고려하여야 한다. 제안한 알고리즘은 이와 같은 역작용을 방지할 수 있는 방안을 마련하였다.

## 4. 수치 예제 및 실험

### 4.1 수치 예제

작업에 대한 자원의 재협상 알고리즘은 대체 공정상의 모든 자원에 대하여 EFT를 만족하는 자원이 있는 경우와 없는 경우가 있다. 만족하지 못하는 경우에 자원간의 재협상 알고리즘이 유용하다. 표 4.1은 예제에서 현재 할당된 작업 상태를 나타낸다.

어떤 작업 A의 Task announcement의 내용은 (part 10, Operation 1, Processing time=5, Due\_date=12) 이고, 작업 복잡도(criticality)는 Processing time\*1.2로 가정한다. 재협상 알고리즘 각 단계의 결과는 다음과 같다.

단계 1)  $EFT(\text{Resource } 1) = ERT(\text{Resource } 1) + t_{\text{ann}} = 11 + 5 = 16$ ,  $EFT(\text{Resource } 2) = ERT(\text{Resource } 2) + t_{\text{ann}} = 9 + 5 = 14$ ,  $EFT(\text{Resource } 3) = ERT(\text{Resource } 3) + t_{\text{ann}} = 9 + 5 = 14$ . 3개의 대체가능한 자원 모두가  $EFT > \text{납기}$  이므로 단계 2로 간다.

단계 2)  $W(\text{Resource } 1) = 11 - 5 = 6$ ,  $W(\text{Resource } 2) = 9 - 2 = 7$ ,  $W(\text{Resource } 3) = 9 - 5 = 4$ ,

단계 3) W의 오름차순은 Resource 3(4) - Resource 1(6) - Resource 2(7)의 순이다.  $t_j$ 의 오름차순은 Resource 2(2) - Resource 1(5), Resource 3(5)의 순이다.  $X = \text{Resource } 3$ ,  $Y = (\text{part } 6, \text{operation } 1)$

단계 4) 물리적으로 작업가능하므로 단계 7로 간다.

단계 7)  $Z = 4 + 2 + 0 + 5 = 11$ . 납기를 만족한다. 그러므로 단계 8로 간다.

단계 8) 예제의 간편성을 위해 X의 criticality와 Y의 criticality이 모두 만족한다고 가정하고 두 작업의 교환을 인정하고 재협상을 마친다.

표 4.1 현재 할당된 작업 상태

대체 가능한 자원	대기중인 작업	부품 #	Processing time	Due_date
Resource 1	Operation 3	Part 1	6	10
	Operation 2	Part 4	5	13
Resource 2	Operation 2	Part 2	3	5
	Operation 3	Part 3	4	10
Resource 3	Operation 1	Part 6	2	15
	Operation 2	Part 5	4	5
	Operation 1	Part 7	5	11

납기를 준수할 수 없었던 Part 8의 Operation이 알고리즘 수행후 납기를 준수하도록 재협상이 이루어진 것을 알 수 있다. 표 4.2는 재협상 알고리즘 수행후의 결과를 나타낸다.

### 4.2 실험 및 결과

기존의 EFT와 본 논문에서 제안한 알고리즘의 수행도를 비교하기 위해 시뮬레이션을 이용하여 실험하는 것이 필요하다. 본 논문에서는 생산현장제어 시스템의 성능에 영향을 줄 수 있는 변수로 공정의 복잡도, 그리고 각 주문에 할당된 납기를 설정하고 납기를 준수하는 것과 계획했던 전체 부품을 최단 시간에 생산하는 것이 중요한 관심사가 된다. 따라서 본 실험에서는 성능 평가를 위한 수행도로 number of tardy jobs, total tardiness 등을 채택하였다. 또한 공정의 복잡도를 고려하여 공정이 간단한 것부터 복잡한 것까지를 4종류로 분류한 자료를 입력하여 생산현장의 복잡도를 변화시켜가며 기존의 EFT(Earlist Finishing Time) 방법과 제안한 알고리즘 사이의 수행도를 비교 분석하였다.



표 4.2 재 협상 알고리즘 수행후의 결과

대체 가능한 자원	대기중인 작업	부품 #	Processing time	Due_date
Resource 1	Operation 3	Part 1	6	10
	Operation 2	Part 4	5	13
Resource 2	Operation 2	Part 2	3	5
	Operation 3	Part 3	4	10
	Operation 1	Part 7	5	11
Resource 3	Operation 2	Part 5	4	5
	Operation 1	Part 6	2	15
	Operation 1	Part 8	5	12

입력자료는 난수를 이용하여 공정의 갯수가 5개인 것, 10개인 것, 15개인 것, 그리고 20개 이상인 것 등 4종류로 나누어서 발생시켰다. 각 주문에 대한 공정의 발생은 U(1,20)을 이용하여 발생시켰다. 각 공정별 평균 가공시간  $\mu$ 은 U(100, 200)을 이용하여 발생시켰고, 최종 가공시간은 Trangular( $0.75 \mu$ , ,  $1.25 \mu$ )로부터 발생시켰다. 시뮬레이션은 4종류의 data set에 대하여 이루어졌으며, 각 data set에 대하여 매회 40개의 제품을 생산하도록 하였다. 1simulation 단위 시간을 현실세계의 5초로 나타내어 100,000 simulation 단위시간동안 실험을 실시하였다. 하루 24시간 가동을 기준으로 할 때, 100,000 단위시간은 실제 시간 6일에 해당한다. 표 4.3부터 표 4.10에서는 공정 갯수와 납기를 변화시켜가며 실험한 결과가 나와 있다. 납기는 각 작업의 공정시간의 합과 납기사이의 비율에 따라 구분한 것으로서, 각각 가공시간의 1.2배, 1.5배, 1.8배, 2.2배의 값을 할당하였다.

표 4.3에서 표 4.10을 통하여 제안된 알고리즘이 기존 EFT 알고리즘 보다 우월함을 알 수 있다. 제안된 알고리즘은 Number of tardy jobs, Total tardiness 등의 성능평가 척도에서 좋은 성능을 나타내 주고 있으며 이는 효율적인 자원할당에 의한 것으로 해석할 수 있으며 자원이용의 극대화, 재협상에 의한 자원들의 가상적인 용량 증대 등에 의하여 가능한 것으로 분석된다. 작업당 공정의 갯수가 많아질수록 성능 차이는 더욱 커지고 있다. 이와 같은 현상은 재협상에 기인한 것으로 처리해야 할 공정의 갯수가 많아질 때, 재협상을 통하여, 다른 자원의 유휴 용량을 마치 자기의 용량인 것처럼 이용함으로써 전체적으로 시간당 시스템의 가용자원이 증가하기 때문이다.

### 5. 결 론

본 연구는 동적인 환경변화에 효율적으로 대처할 수 있는 홀론 개념을 이용하여 동적인 주문에 대해 주문 신뢰성 및 유연성을 제공할 수 있는 자원 할당 알고리즘에 관하여 연구하였다. 결과로 보다 동적인 자원 할당과 진단을 포함한 확장된 홀로닉 자원 관리 시스템의 개발을 위한 모델링 및 운영기법의 토대를 마련하였다. 또한 작업중심의 생산일정과 자원 측면의 생산 일정을 모두 고려한 협상 및 재협상 알고리즘을 개발하였다.

추후 연구 과제로 대체 가능한 자원이 서로 다른 목적함수를 갖는 경우를 고려한 자원할당 알고리즘 개발이 필요하고 또한 다양한 이상상황을 고려한 홀로닉 자원관리시스템 모델의 개발이 요구된다.

참고 문헌

- [1] 이영해, 김정, 홀로닉 생산시스템의 개요 및 연구 방향, 제어자동화시스템공학회지, 제3권 제1호, pp.33-40, 1997
- [2] 문장석, 박진우, 장성용, 자치적 객체들간의 Bidding을 통한 FMS생산 현장 통제에 관한 연구, 93추계학술발표대회 논문집, 대한산업공학회, pp.38-47, 1993.
- [3] 이용수, 이영해, 전성진, Holonic 생산시스템의 지능제어모델, 한국경영과학회/대한산업공학회, 95춘계공동학술대회 논문집, pp 47-55, 1995.
- [4] Agre, J.R., G. Elsley, D. McFarlane, J. Cheng, and B. Gunn, "Holonc Control of a Water Cooling System for a Steel Rod Mill", Proceedings of the 4th International Conference on CIM and Automation Technology, New York, pp.134-141, Oct. 1994.
- [5] Davis, R., and R. G. Smith, Negotiation as a Metaphor for Distributed Problem Solving, Artificial Intelligence, Vol.20, 1983
- [6] Park, N., and J. W. Park, A Cooperative-distributed Shop Floor Control for Computer Integrated Manufacturing through Negotiation and Task Sharing, Proceedings of the 2nd International Conference on Decision Science, Seoul, 1993
- [7] Shaw, M. J., Dynamic Scheduling in Cellular Manufacturing System: A Framwork for Networked Decision Making, Journal of Manufacturing Systems, Vol.7, No.2, pp.83-94, 1988
- [8] Tonshoff, H. K., M. Winkler, and J.C. Aurich, "Product Modeling for Holonic Manufacturing Systems", Proceedings of 4th International Conference on CIM and Automation Technology, New York, pp.121-127, October 1994.
- [9] Valckenaers, P., F. Bonneville, H. V. Brussel, L. Bongaerts, and J. Wyns, "Results of the Holonic Control System Benchmark at KULeuven", Proceedings of the 4th International Conference on CIM and Automation Technology, New York, pp.128-133, Oct. 1994.
- [10] Valckenaers, P., and H. Van Brussel, Holonic Manufacturing Systems: Technical Overview, Technical Report, Katholieke Univ. Leuven, Division PMA Belgium, Dec. 1995.
- [11] Wyns, J., P. Valckenaers, and H. V. Brussel, "Resource Allocation in the Holonic Manufacturing System at K.U.Leuven", Proceedings of Conference on Control of Industrial System, Belfort, France, May 1997.

표 4.3 data set 1에 대한 Number of tardy jobs

		EFT				제한한 알고리즘			
		공정 개수				공정 갯수			
		5개	10개	15개	20개	5개	10개	15개	20개
납 기	가공시간x1.2	6.5	5.2	8.2	10.7	5.9	5.5	8.5	12.0
	가공시간x1.5	2.7	4.1	5.2	6.9	2.0	3.3	3.6	3.9
	가공시간x1.8	2.3	2.3	3.6	4.7	1.5	1.4	1.8	2.3
	가공시간x2.2	2.2	2.9	4.2	4.8	2.1	2.5	3.6	4.8

표 4.4 data set 2에 대한 Number of tardy jobs

		EFT				제한한 알고리즘			
		공정 개수				공정 갯수			
		5개	10개	15개	20개	5개	10개	15개	20개
납 기	가공시간x1.2	3.3	6.2	9.3	11.4	3.4	5.9	9.7	12.3
	가공시간x1.5	2.2	3.5	4.7	6.1	2.0	3.2	5.0	4.2
	가공시간x1.8	1.0	2.3	3.5	3.8	1.0	1.8	1.8	2.3
	가공시간x2.2	1.8	2.6	3.6	3.9	2.0	2.2	3.0	3.8

표 4.5 data set 3에 대한 Number of tardy jobs

		EFT				제한한 알고리즘			
		공정 개수				공정 갯수			
		5개	10개	15개	20개	5개	10개	15개	20개
납 기	가공시간x1.2	5.2	7.3	10.4	13.2	4.8	7.5	9.8	14.5
	가공시간x1.5	3.2	3.5	5.3	6.7	2.5	2.6	2.9	3.3
	가공시간x1.8	2.1	2.6	3.5	4.7	1.8	1.9	1.6	1.9
	가공시간x2.2	2.0	2.4	3.2	4.2	2.0	2.3	2.8	3.9

표 4.6 data set 4에 대한 Number of tardy jobs

		EFT				제한한 알고리즘			
		공정 개수				공정 갯수			
		5개	10개	15개	20개	5개	10개	15개	20개
납 기	가공시간x1.2	4.3	3.2	4.5	4.7	5.0	3.2	3.9	4.8
	가공시간x1.5	2.4	2.5	2.5	6.5	2.2	2.4	2.3	5.0
	가공시간x1.8	1.3	1.5	2.1	1.9	1.1	1.2	1.3	1.5
	가공시간x2.2	2.3	2.2	2.5	2.9	2.2	2.2	2.3	2.7

표 4.7 data set 1에 대한 Total tardiness

		EFT				제안한 알고리즘			
		공정 갯수				공정 갯수			
		5개	10개	15개	20개	5개	10개	15개	20개
납 기	가공시간x1.2	28758	30124	31527	34172	28915	29207	30508	33851
	가공시간x1.5	26123	27710	29316	30241	25416	26120	27065	35442
	가공시간x1.8	11824	13185	25903	32905	9902	11786	15629	22154
	가공시간x2.2	25421	26825	28397	31092	24837	26513	27924	30510

표 4.8 data set 2에 대한 Total tardiness

		EFT				제안한 알고리즘			
		공정 갯수				공정 갯수			
		5개	10개	15개	20개	5개	10개	15개	20개
납 기	가공시간x1.2	28507	29351	36163	38433	27793	30367	36098	38192
	가공시간x1.5	24615	28333	31285	33706	22106	26508	28631	30359
	가공시간x1.8	21153	27854	29037	33251	16025	17203	17460	20147
	가공시간x2.2	24239	27894	30406	32985	22994	21297	26223	29600

표 4.9 data set 3에 대한 Total tardiness

		EFT				제안한 알고리즘			
		공정 갯수				공정 갯수			
		5개	10개	15개	20개	5개	10개	15개	20개
납 기	가공시간x1.2	25201	28637	30295	37264	25638	28105	29672	35614
	가공시간x1.5	24320	26023	29549	35925	23364	24603	30238	32589
	가공시간x1.8	20154	25960	31259	34682	19053	21150	26583	28576
	가공시간x2.2	23170	23051	29605	32247	22854	23429	26034	30514

표 4.10 data set 4에 대한 Total tardiness

		EFT				제안한 알고리즘			
		공정 갯수				공정 갯수			
		5개	10개	15개	20개	5개	10개	15개	20개
납 기	가공시간x1.2	26357	29349	32185	35127	27462	28639	32204	35201
	가공시간x1.5	23982	26160	29308	33105	23017	24139	26315	29057
	가공시간x1.8	22165	26512	29332	31894	21953	22876	25497	29034
	가공시간x2.2	23857	24593	26871	28353	22856	24037	25452	26341