

# 다중기계로 구성되는 조립전단계에서의 부품생산 일정계획

윤상 흠

전주대학교 정보기술컴퓨터공학부

## Scheduling on the Pre-assembly Stage of Multiple Fabrication Machines

Sang-Hum Yoon

School of Information Technology and Computer Engineering, Jeonju University

This paper analyses a deterministic scheduling problem concerned with manufacturing multiple types of components at a pre-assembly stage composed of parallel fabrication machines. Each component part is machined on a fabrication machine specified in advance. The manufactured components are subsequently assembled into products. The completion time of a job(product) is measured by the latest completion time of its all components at the pre-assembly stage. The problem has the objective measure of minimizing the total weighted completion time of a finite number of jobs. Two lower bounds are derived and tested in a branch-and-bound scheme. Also, three constructive heuristic algorithms are developed based on the machine aggregation and greedy strategies. Some empirical evaluation of the performance of the proposed branch-and-bound and heuristic algorithms are also performed.

**Keywords :** Fabrication Scheduling, Branch-and-Bound, Heuristics

### 1. 서 론

일반적으로 조립라인은 작업흐름에 따라 두 가지로 구분될 수 있다. 첫 번째는 잘 알려진 흐름생산라인(flowshop)으로 반제품이 물류장비를 통해 이동하면서 직렬상의 각 설비를 통해 해당 부품이 조립됨으로써 제품이 점차 완성되는 형태이다. 두 번째는 최종조립에 필요한 각 부품(구성품, component)이 서로 다른 설비와 경로(route)를 통해 독립적으로 생산되고, 완성된 부품들은 조립단계로 이동한 후 최종 조립공정을 통해 제품으로 완성되는 형태이다. 이러한 생산체계에서는 조립전단계(pre-assembly stage)에 위치한 각 부품 생산설비와 조립단계간에 물리적으로 2단계 직렬구조를 가지지만 부품 생산설비들간에는 독립적인 병렬구조를 가진다. 이를 일반적으로 AFS(assembly-type flowshop)라고 부르고 있으며(Lee et al.[8], Sun et al.[11]), 본 논문에서는 AFS의 조립전단계를 대상으로 부품생산을 위한 일정계획문제를 고려하고 있다.

AFS는 흐름생산라인에 비해서 부품의 개수가 작고, 부품과 완제품의 크기가 큰 경우에 적절하다. 예를 들어, Lee et al.[8]의 연구에서 지적되었듯이 엔진의 구성품인 본체(body)와 쇄시(chassis)는 서로 다른 공정이나 플랜트에서 생산된 후 고정된 위치의 조립단계로 이동한 후 주조립공정을 통해 엔진으로 완성된다. Sun et al.[11]은 여러 구성품들이 작은 로트단위의 단품종으로 완성되는 FMC(flexible manufacturing cell)를 AFS로 모델링하고 연구한 바 있다. 또한, 병렬처리시스템에서도 큰 프로그램을 작은 단위의 단위프로그램으로 분리하고 각각의 단위프로그램이 독립적으로 수행된 후 이를 메인 프로세서에서 병합하는 과정을 거치게 되며 대기행렬이론에서는 이러한 분리 및 병합구조를 포크/조인(fork/join) 시스템이라 한다(Heidelberger&Trivedi[7]).

AFS에서는 제품설계단계에서 주조립공정을 단순화하기 위해 부품위주설계(fabrication-driven design) 또는 모듈러설계(modular design)를 채택할 수 있으며, 이 경우 부품생산공정이 조립공정에 비해 보다 복잡해지고 긴

가공시간이 소요된다(He&Kusiak[6]). 더욱이 Baker et al.[1]는 확률과정을 활용한 연구를 통해 AFS의 생산율(throughput)을 높이기 위해서는 라인의 설계단계에서 조립공정에 비해 부품생산공정에 상대적으로 높은 작업부하(workload)가 부과되도록 비대칭 설계를 하는 것이 유리하다는 연구결과를 제시한 바 있다. 이는 조립전단계가 조립단계에 비해 상대적으로 병목(bottleneck)공정이 됨을 의미하게 되며 일정계획이론의 관점에서 볼 때 전체 조립라인의 성능향상을 위해서는 조립전단계의 부품생산 일정계획이 중요한 과제가 됨을 의미한다.

따라서 본 논문에서는 최종제품조립에 필요한 부품들이 조립전단계의 병렬기계를 통해 독립적으로 생산되는 경우에 제품조립을 효율적으로 지원할 수 있는 부품생산을 위한 일정계획을 다루고 있다. 제품은 여러 개의 부품으로 구성되고 각 부품은 미리 지정된 기계를 통해 생산된다. 최종 조립공정을 위해서는 필요한 모든 부품의 생산이 완료된 시점에서 가능하므로 본 연구에서는 각 작업(제품)의 완료시점을 전체 부품이 생산완료되는 시점으로 모델링하였다. 본 연구의 목적은 전체 작업에 대한 총가중완료시간(total weighted completion time)이 최소화되는 부품생산 일정계획을 도출하는 것이다.

부품생산을 위한 일정계획과 관련하여 최근에 이르기 까지 활발한 연구가 진행되고 있다. Lee et al.[8]은 두 대의 부품생산기계와 조립기계가 연결된 2단계 AFS에서의 일정계획문제를 최초로 소개하였다. Potts et al.[10]은 이 문제를 부품생산기계가 여러대인 경우로 확장하였으며, Sun et al.[11]은 같은 문제에 대해 다양한 발견적 해법을 제시하였다. 이들의 연구는 조립전단계뿐 아니라 조립단계까지 고려하여 본 논문에 비해 물리적으로 확장된 시스템을 대상으로 하고 있으나, 일정계획을 위한 목적함수를 최대 작업완료시간(makespan)으로 한정하여 부품에 대한 재공품재고의 최소화나 작업별 중요도를 고려할 수는 없다. 본 논문에서는 이러한 측면을 고려하기 위해 목적함수를 총 가중완료시간으로 채택하고 있다.

Lin[9], Sung&Park[12], Cheng&Wang[2], Coffman et al.[3], Gerodimos et al.[5] 등은 단일설비에서 모든 부품이 생산되는 경우를 대상으로 연구를 수행하였다.

본 논문에서와 같이 병렬기계를 가지는 조립전단계에 대한 일정계획문제는 Sung&Yoon[13]에서 제시되었다. 그들은 부품생산설비가 두 대인 경우를 대상으로 하여 총가중완료시간을 목적함수로 할 경우 일정계획문제가 NP-난도(NP-complete)문제임을 증명하고 몇가지 발견적 해법을 제시하였다. 본 논문에서는 이러한 선행연구를 설비가 여러대인 경우로 일반화하고 최적해를 위한 효율적인 분지한계법(branch and bound)과 효과적인 세가지

발견적 해법을 제시한다.

논문의 구성은 다음과 같다. 2절에서는 문제의 정확한 이해를 돋기 위해 고려하고 있는 일정계획문제에 대한 수리적 정의를 하고 3절에서는 두 가지 하한(lower bound)도출을 통한 분지한계법을 제시하였다. 4절에서는 기계병합과 탐색기법에 근거한 세 가지 발견적 해법을 제안하였다. 5절에서는 다양한 수치실험을 통해 제안된 분지한계법과 발견적해법에 대한 성능분석을 수행하였다. 마지막으로 6절에서는 결론과 추후연구방향에 대한 논의가 이루어진다.

## 2. 문제정의 및 분석

본 논문에서는  $m$ 개의 병렬기계  $M_1, M_2, \dots, M_m$ 로 구성되는 작업장에서  $n$ 개의 작업  $J_1, J_2, \dots, J_n$ 을 수행하는데 필요한 일정계획 수립을 대상으로 한다. 각 작업은  $m$ 개의 부분작업(sub-task)들로 구성되며 각 부분작업의 수행은 미리 지정된 기계에서 이루어진다. 즉, 각 작업  $J_i$ 의  $k$ 번째( $k = 1, 2, \dots, m$ ) 부분작업은 기계  $M_k$ 에서 수행되며  $p_{i,k}$ 의 작업시간이 소요된다. 각 작업의 작업완료시간은 해당 부분작업들이 모두 완료되는 시점으로 모델링된다. 따라서, 일정계획  $S$ 가 주어졌을 때  $c_{i,k}(S)$ 를 작업  $J_i$ 의  $k$ 번째 부분작업이 기계  $M_k$ 에서 완료되는 시점이라고 한다면 작업  $J_i$ 의 작업완료시간  $C_i(S)$ 는 다음과 같이 계산될 수 있다.

$$C_i(S) = \max \{ c_{i,k}(S) \mid k = 1, 2, \dots, m \}$$

또한, 각 작업에는 작업완료시간에 대한 가중치  $w_i$ 가 미리 할당되어 전체 작업에 대한 총 가중완료시간은 다음과 같이 표현된다.

$$\begin{aligned} TC_w(S) &= \sum_{i=1}^n w_i C_i(S) \\ &= \sum_{i=1}^n w_i \max \{ c_{i,k}(S) \mid k = 1, 2, \dots, m \} \end{aligned}$$

본 논문에서 다루고 있는 일정계획은  $TC_w(S)$ 를 최소화하는 각 기계에서의 부분작업에 대한 가공순서  $S$ 를 결정하는 것이다. 일정계획에서 각 작업은 고객으로부터의 주문을 의미하며, 가중치  $w_i$ 는 개별주문에 대한 중요도를 나타내는 지표로 해석될 수 있다. 또한, 본 문제에서는 각 기계가 다른 기계와 독립적으로 작업을 수행함으로 임의의 지연시간(idle time)이 필요하지 않음을

주목할 필요가 있다. 따라서, 모든 작업이 완료되는 최종 완료시점은 다음과 같이 항상 고정된 상수값을 가지게 된다.

$$C_{\max} = \max \left\{ \sum_{i=1}^n p_{i,k} \mid k=1, 2, \dots, m \right\}$$

이때, 모든 작업에 대해 가중치가  $w_i=1$ 인 경우에는 작업장에 머무르는 평균 부분작업수( $N(S)$ )와 총 가중완료시간과의 다음과 같은 관계식을 통해(Conway et al.[4]) 본 논문에서 다루고 있는 일정계획문제가 작업장의 제공품재고를 최소화하게 되는 효과가 있음을 알 수 있다.

$$N(S)/(n \cdot m) = TC_w(S)/C_{\max}$$

다음의 정리는 주어진 문제의 최적일정계획이 모든 기계에서 부분작업들이 동일한 순서로 수행되는 순열일정계획(permutation schedule)에 의해 성격지워짐을 증명한 것이다.

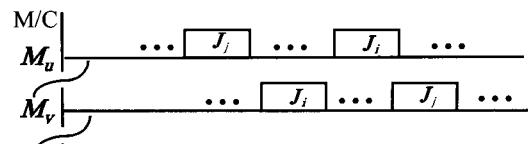
**정리 1.** 최적일정계획은 순열일정계획에서 존재한다.

**증명.** 증명은 어떠한 비순열일정계획(non-permutation schedule)도 작업교환과정을 통해 더 좋거나 최소한 같은 목적함수값을 가지는 순열일정계획으로 항상 변환될 수 있음을 보이게 된다. 임의의 비순열일정계획  $S$ 를 고려하자. 이때, 두 개의 작업  $J_i$  와  $J_j$ 의 처리순서가 임의의 두 기계  $M_u$ 와  $M_v$ 에서 서로 다르다고 가정하자. 또한,  $M_u$ 에서 두 작업이 모두 완료되는 시점이  $M_v$ 에서의 완료시점보다 작거나 같다고 하면 <그림 1>에서와 같이 두 가지 경우가 나타날 수 있다. 이때 각 경우에 대해 적절한 작업교환을 통해 모든 작업에 대해 완료시간이 증가하지 않으면서 두 작업의 처리순서를 두 기계에 동일하게 만들 수 있다.

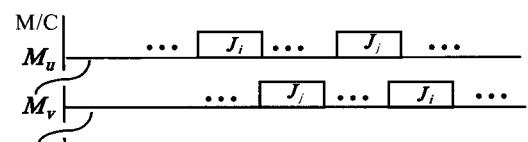
<그림 1.1>의 경우:  $M_u$ 에서  $J_j$  가  $J_i$  보다 처리 순서상 앞선 경우에는  $M_u$ 에서  $J_j$  를  $J_i$  의 바로 다음 위치로 이동하여 삽입한다.

<그림 1.2>의 경우:  $M_u$ 에서  $J_i$  가  $J_j$  보다 앞선 경우에는  $M_u$ 에서  $J_i$  를  $J_j$  의 바로 다음 위치로 삽입한다.

이와 같은 작업교환을 처리순서가 다른 모든 작업쌍을 찾아서 반복수행하게 되면 항상 순열일정계획을 찾을 수 있다.



(경우 1)



(경우 2)

<그림 1> 비순열일정계획의 예

### 3. 분지한계법(branch-and-bound)

#### 3.1. 분지전략(branching rule)

정리 1에 의해서 주어진 문제는  $n!$  개의 가능한 모든 순열일정계획에서 최적의 순열을 찾는 것과 같은 문제가 된다. 따라서 분지한계트리의 각 노드(node)는 전체 작업들 중 일부 작업들로 구성되는 부분순열(partial sequence)에 일대일로 대응된다. 이러한 부분순열은 전체순열의 앞부분에 위치하게 된다.

분지단계에서 분지할 하나의 노드(부분순열)를 선택하고 이 부분순열에 포함되지 않은 작업들 중 한개의 작업을 이 부분순열의 바로 뒤에 삽입함으로써 새로운 자식노드가 발생된다. 따라서, 만약 어떤 노드가  $a$ 개의 작업들로 구성되는 부분순열에 해당한다면 이 노드를 분지함으로써  $(n-a)$ 개의 자식노드가 발생된다.

다음으로 분지할 노드의 선택은 깊이우선탐색(depth-first search)을 원칙으로 한다. 이는 탐색트리에서 깊이내려갈수록 부분순열의 작업수가 커지므로 가장 큰 부분순열을 가진 노드가 선택됨을 의미한다. 이러한 노드가 여러개 발생할 경우에는 그중에서 하한값(lower bound)이 가장 작은 노드가 선택된다.

#### 3.2. 한계전략(bounding rule)

여기서는 분지한계법에 사용될 두 개의 하한식을 도출한다. 이를 위해 다음과 같은 기호를 사용한다.

- |           |                                  |
|-----------|----------------------------------|
| $\sigma$  | = 작업들의 부분집합으로 구성되는 부분 순열         |
| $\sigma'$ | = $\sigma$ 에 포함되지 않은 나머지 작업들의 집합 |

$n_{\sigma}$  =  $\sigma$ 에 포함된 작업의 수

$TC_w(\sigma)$  =  $\sigma$ 에 대한 총 가중완료시간

$t_k(\sigma)$  =  $\sigma$ 에 대한 기계  $M_k$ 에서의 최종 완료시간

따라서,  $\sigma$ 로 시작되는 어떤 일정계획  $S$ 에 대한 총 가중완료시간은

$$TC_w(S) = TC_w(\sigma) + TC_w(\sigma') \dots (1)$$

와 같이 표현할 수 있다. 여기서 편의를 위해 작업을 나타내는 기호(index)가  $S$  상의 순서에 따라 정렬되어 있으며 이후의 수식에서  $\max$  연산은 항상 기계를 표시하는 기호  $k$ 에 대해서 적용된다고 하자.

식 (1)에서  $TC_w(\sigma')$ 에 대해 다음과 같은 전개가 성립하게 된다.

$$\begin{aligned} TC_w(\sigma') &= \sum_{i \in \sigma'} w_i C_i(S) \\ &= \sum_{i \in \sigma'} w_i \max \left\{ t_k(\sigma) + \sum_{j=n_{\sigma}+1}^i p_{j,k} \right\} \\ &\geq \max \left\{ \sum_{i \in \sigma'} w_i \left( t_k(\sigma) + \sum_{j=n_{\sigma}+1}^i p_{j,k} \right) \right\} \\ &= \max \left\{ t_k(\sigma) \sum_{i \in \sigma'} w_i + \sum_{j=n_{\sigma}+1}^n w_i \sum_{k=j+1}^i p_{j,k} \right\} \end{aligned} \quad \dots (2)$$

이때 식(2)의 마지막 식에서  $\sum_{j=n_{\sigma}+1}^n w_i \sum_{k=j+1}^i p_{j,k}$ 는 단일기계 일정계획문제에서  $\sigma'$ 에 속한 작업들의 총 가중완료시간과 같음을 알 수 있다. 따라서,  $WSPT_k(\sigma')$ 가  $n_{\sigma'}$ 개의 작업집합  $\{p_{i,k}, w_i \mid i \in \sigma'\}$ 에 대한 단일기계일정계획 문제에서 작업순서를  $WSPT$  (weighted shortest processing time)순서로 처리할 경우 발생되는 총 가중완료시간을 나타낸다고 할 때 다음의 식이 성립한다.

$$TC_w(\sigma') \geq \max \{ t_k(\sigma) \sum_{i \in \sigma'} w_i + WSPT_k(\sigma') \}$$

따라서, 분지한계법의 어떤 노드에서 부분순열  $\sigma$ 로 시작되는 전체 일정계획에 대한 하한값은 식(1)에 의해 다음과 같이 계산된다.

$$\begin{aligned} LB_1 &= TC_w(\sigma) \\ &+ \max \{ t_k(\sigma) \sum_{i \in \sigma} w_i + WSPT_k(\sigma') \} \end{aligned}$$

각 노드에서  $LB_1$ 을 계산하기 위한 시간복잡도(time complexity)는  $O(m \cdot n_{\sigma} \cdot \log n_{\sigma})$ 이다.

두 번째 하한값은 (2)의 두 번째 식으로부터 시작하여 다음의 전개를 통해 도출될 수 있다.

$$\begin{aligned} (2) &\geq \sum_{i \in \sigma'} w_i \sum_{k=1}^m \left( t_k(\sigma) + \sum_{j=n_{\sigma}+1}^i p_{j,k} \right) / m \\ &= \left( \sum_{i \in \sigma'} w_i \sum_{k=1}^m t_k(\sigma) + \sum_{i \in \sigma'} w_i \sum_{k=1}^m \sum_{j=n_{\sigma}+1}^i p_{j,k} \right) / m \end{aligned}$$

이때 마지막 식에 포함된  $\sum_{i \in \sigma'} w_i \sum_{k=1}^m \sum_{j=n_{\sigma}+1}^i p_{j,k}$ 는  $\left\{ \sum_{k=1}^m p_{i,k}, w_i \mid i \in \sigma' \right\}$ 의 값을 가지는  $n_{\sigma'}$ 개의 작업집합에 대한 단일기계일정계획문제의 총가중작업완료시간을 나타내고 있음을 알 수 있다. 이를 활용하면 다음과 같이 두 번째 하한값을 도출할 수 있다.

$$\begin{aligned} LB_2 &= TC_w(\sigma) \\ &+ \sum_{i \in \sigma'} w_i \sum_{k=1}^m t_k(\sigma) / m + WSPT_{\text{sum}}(\sigma') / m \end{aligned}$$

여기서,  $WSPT_{\text{sum}}(\sigma')$ 는 언급된 단일기계 일정계획 문제에서  $WSPT$  순서를 사용했을 경우의 총가중작업완료시간을 나타낸다.  $LB_2$ 를 계산하기 위해 소요되는 계산량은  $O(n_{\sigma'} \log n_{\sigma'})$ 이다.

따라서 본 논문의 분지한계법에서 사용되는 각 노드의 최종적인 하한값은  $LB_1$ 과  $LB_2$ 중에 큰 값을 사용하게 된다.

$$LB = \max \{ LB_1, LB_2 \}$$

### 3.3. 초기 상한값(upper bound)

분지한계법의 효율을 높이기 위해 분지한계트리의 루트노드(root node)는 발견적해법을 통해 발생되는 가용해(feasible solution)의 목적함수값을 전체 문제의 초기 상한값으로 가지게 된다. 만약 분지한계법의 진행과정에서 새롭게 발생된 노드의 하한값이 이러한 상한값보다 클 경우 그 노드는 더 이상 고려될 필요가 없이 제거된다. 본 논문의 분지한계법에서는 다음의 4절에서 소개되는 3가지 발견적 해법의 결과중에서 가장 좋은 해가 루트노드의 초기 상한값을 제공하게 된다.

## 4. 발견적 해법

본 논문에서 제시된 일정계획문제는 Sung&Yoon[13]를

통해 기계가 단지 두 대인 경우( $m=2$ )에도 NP-난도문제임이 증명된 바가 있다. 따라서, 크기가 작은 문제에 대한 최적해 도출뿐 아니라 큰 문제에 대해서는 작은 계산량으로 효과적인 해를 제시할 수 있는 발견적기법의 개발이 중요한 과제라 할 수 있다. 또한 본 절에서 제시된 해법들은 그 자체로써의 중요성 뿐만 아니라 3절에서 제시된 분지한계법에서 루트노드의 상한값을 제공함으로써 탐색량을 크게 줄이는 효과를 제공한다.

처음 두 개의 발견적해법의 아이디어는 전체 병렬기계를 단일기계로 병합하여 문제를 단일기계 일정계획문제로 단순화한 다음 WSPT순서를 적용하여 해를 구하는 것이다. 이때 병합된 가상의 단일기계의 작업시간을 어떻게 정하느냐에 따라 두 가지 해법으로 나뉘게 된다. 첫 번째 방법은 각 작업에 해당하는 부분작업시간들의 최대값을 활용하는 것이다.

#### 발견적 해법 1.(HEU-1)

Step 1. 각 작업  $J_i$ 에 대해 부분작업시간중 최대값  $t_i = \max\{p_{i,k}\}, i=1, 2, \dots, n$ 을 계산한다.

Step 2. 작업집합  $\{t_i, w_i | i=1, 2, \dots, n\}$ 로 구성되는 단일기계 일정계획문제의 WSPT순열을 구하고, 이를 원래 문제의 일정계획으로 사용한다.

부품생산기계에서 생산된 부품들을 최종 조립기계에 적재하는 과정은 Baker et al.[1]의 연구에서 지적된 바와 같이 동시적재(simultaneous loading)와 개별적재(independent loading)로 구분할 수 있다. 개별적재에서는 하나의 완성된 부품은 다른 부품들의 완성여부와 상관없이 조립기계의 입력버퍼로 독립적인 이동이 가능하지만 동시적재 상황에서는 제품조립을 위한 모든 부품이 완성되기 전까지 개별부품이 조립기계에 독립적으로 적재되는 것을 허용하지 않는다. 따라서, 동시적재하에서 부품생산기계에 별도의 출력버퍼가 없는 상황이라면 완성된 부품이 적재될 공간이 없게 되고 부품생산기계에서는 블로킹(blocking)이 발생될 수 있다. 이러한 특수한 상황에서는 제시된 HEU-1이 최적해를 제시할 수 있음을 쉽게 알 수 있다.

두번째 발견적 해법은 단일기계로의 병합시에 각 작업의 작업시간을 전체 부분작업시간의 합으로 상정하는 방법이다.

#### 발견적 해법 2.(HEU-2)

Step 1. 각 작업  $J_i$ 에 대해 부분작업시간들의 합  $t_i = \sum_{k=1}^m p_{i,k}, i=1, 2, \dots, n$ 을 계산한다.

Step 2. HEU-1의 Step 2와 동일한다.

제시된 발견적 해법들의 시간복잡도는  $O(n \log n)$ 이며 매우 빠른 시간에 해를 도출할 수 있다. 다음 예제는 해법에 대한 이해를 돋기 위한 것이다.

예제 1. 4개의 작업을 2대의 병렬기계에서 수행하는 문제에서 부분작업시간과 가중치가  $p_{i,1} = (3, 5, 2, 7), p_{i,2} = (5, 2, 3, 6), w_i = (2, 1, 2, 3)$ 와 같이 주어졌다고 하자. 이때 HEU-1을 통한 일정계획과 목적함수값은  $S = J_3 J_4 J_1 J_2, TC_w(S) = 78$  이고 HEU-2의 결과는  $S = J_3 J_1 J_2 J_4, TC_w(S) = 81$  이다.

마지막으로 보다 효과적인 해의 도출을 위해 탐욕적 탐색(greedy search)을 활용한 발견적 해법을 제시한다. 이 해법에서는 부분순열이 전체순열이 될 때까지 매 단계마다 한 개의 작업을 선택하여 부분순열의 끝에 삽입하게 된다. 삽입작업의 선택은 이미 완성된 부분순열과 새롭게 선택되는 작업을 삽입함으로써 발생되는 목적함수값을 기준으로 이루어진다. 매 단계마다 아직 선택되지 않은 작업들을 탐색하게 되고 총  $n$ 번의 단계가 필요하므로  $O(n^2)$ 의 시간복잡도가 소요된다.

#### 발견적해법 3.(HEU-3)

초기화.  $x = 1, \sigma = \emptyset, \sigma' = \{J_1, J_2, \dots, J_n\}, T = 0$

Stage 1. (첫번째 작업 선택단계).

Step 1. 각 작업  $J_i, i=1, 2, \dots, n$ 에 대해 다음을 계산한다.

$$\max\{p_{i,k} | k=1, 2, \dots, m\}/w_i$$

Step 2. Step 1의 계산결과에서 가장 작은 값을 가지는 작업을  $\sigma$ 의 첫 번째 위치에 삽입하고  $\sigma'$ 에서는 그 작업을 제거한다.

Stage 2. (반복단계)

Step 3. 만약  $x = n$ 이면 Step 4를 수행하고  $x < n$ 이면 Step 5-9를 수행한다.

Step 4. 알고리즘을 종료하고 현재의  $\sigma$ 가 최종해임.

Step 5.  $x = x + 1$

Step 6. 현재의  $\sigma$ 에 속한 작업들의 최종완료시간  $T = \max\{t_k(\sigma) | k=1, 2, \dots, m\}$ 을 계산하고  $T$ 와 각 기계의 완료시간간의 차이를 나타내는  $\Delta_k$ 를 다음과 같이 계산한다.

$$\Delta_k = T - t_k(\sigma), k=1, 2, \dots, m$$

Step 7.  $J_i \in \sigma'$ 에 해당하는 각 작업  $J_i$ 에 대해 다음과 같이  $d_i$ 를 계산한다.

$$d_i = \max\{p_{i,k} - \Delta_k | k=1, 2, \dots, m\}/w_i$$

Step 8. 계산된  $d_i$ 값이 최소인 작업을 선택하여  $\sigma'$ 의  $x$  번째 위치에 추가하고  $\sigma'$ 에서는 그 작업을 제거한다.

Step 9. 현재의  $\sigma'$ 를 수행하기 위한 각 기계의 종료 시간  $t_k(\sigma')$ ,  $k=1, 2, \dots, m$ 를 갱신한다.

다음 예제는 HEU-3의 이해를 돋기 위한 것이다.

예제 2. 예제 1의 문제를 다시 고려하자. Stage 1에서  $J_3$ 이 선택된다. Stage 2에서  $d_4=2$ 의 값으로 최소가 되어  $J_4$ 가 선택된다. 계속된 Stage 3와 4에서  $J_1$ 과  $J_2$ 가 차례로 선택되어 HEU-3에 의해 결정된 최종 일정계획은  $S=J_3J_4J_1J_2$ 이고 목적함수값은  $TC_w(S)=78$ 가 된다.

## 5. 수치실험

### 5.1. 실험설계

본 절에서는 제시된 발견적해법과 분지한계법에 대한 성능평가결과를 수치실험을 통해 보여준다. 실험은 SUN-SPARC Unix 환경에서 수행되었으며 프로그램은 C언어를 사용하였다.

실험에 사용된 문제는 작업수가 10개에서 100개 사이의 경우를 고려하였다. 가중치  $w_i$ 의 값은 [1,10]의 범위를 가지는 균등분포로부터 무작위로 추출되었다.

본 논문에서 다루고 있는 일정계획은 각 기계에 대한 부하균형(load balancing)여부에 따라 문제의 난이도가 달라진다. 이에 따라 문제유형을 3가지 종류로 나누었다. 문제그룹 1에서는 모든 기계에서의 작업시간을 동일하게 [1,30]의 범위를 가지는 균등분포로부터 추출하였으며 문제그룹 2에서는 첫 번째 기계에서의 작업시간은 [5,30]의 범위에서, 나머지 기계에 대해서는 [1,25]의 범위를 가지게 하였다. 이는 첫 번째 기계가 나머지 기계에 비해 상대적으로 작업부하가 큰 경우를 나타낸 것으로 문제그룹 2가 문제그룹 1에 비해 난이도가 낮게 됨을 의미한다. 이러한 경향을 좀더 확대하여 문제그룹 3에서는 첫 번째 기계에 대해서는 [10,30]의 범위에서 나머지 기계는 [1,20]의 범위에서 작업시간을 추출하였다.

기계수는 2대, 3대, 5대를 고려하였다. 작업수, 문제그룹, 기계수에 따른 각각의 조합에 대해 20문제씩을 생성하여 관련 통계량을 추출하였다.

### 5.2. 성능평가

<표 1>은 분지한계법에 대한 실험결과를 나타낸 것이다. 표에서는 문제를 풀기 위해 사용된 CPU시간에 대한 평균과 중앙값(median)을 보여주고 있다. 또한, 분지한계트리에서 발생될 수 있는 최대 노드수 대비 실제 발생된 노드수에 대한 발생노드비율을 보여주고 있다. 작업수가  $n$ 인 경우 최대 발생가능한 노드의 수는  $1 + n + n(n-1) + \dots + n!$ 가 된다. 따라서 [ $m=2$ ,  $n=10$ , 문제그룹 1]의 경우에 최대 발생가능한 노드의 수는 6,235,301개이고 실제 발생된 평균노드수가 528개라면 노드발생비율은  $100(528/6,235,301) = 8.47E-3\%$ 가 된다. 표에서 볼 수 있듯이 문제사이즈가 커짐에 따라 이러한 비율값은 지속적으로 감소하고 있음을 알 수 있고, 이는 제안된 분지한계법이 효과적으로 기능하고 있음을 나타낸다.

<표 2>와 <표 3>에서는 제안된 발견적해법에 대한 성능평가 결과를 보여주고 있다. 먼저 최적해와의 비교를 위해 작업수가 10개에서 18개 사이에서 다음과 같은 상대오차(relative error)값을 계산하였다.

$$RE = (\text{Heu}-\text{Opt})/\text{Opt}$$

여기서 Heu는 발견적해법에 의한 목적함수값을 의미하고, Opt는 분지한계법에 의해 구해진 최적해의 목적함수값을 의미한다. <표 2>에서는 각 문제조합에 해당하는 20문제씩에 대해 평균상대오차(Mean RE), 최대상대오차(Max. RE), 발견적해법이 최적해를 찾은 문제의 수(Nof)를 보여주고 있다. 3개의 발견적 해법이 모두 우수한 해를 도출함을 알 수 있으며 특히, HEU-3의 경우 Nof의 값이 매우 우수하며 기계수나 작업수가 증가하더라도 해가 크게 나빠지지 않음을 알 수 있다.

다음으로 작업수가 큰 문제에 대해 HEU-3에 대한 실험을 진행하였다. 여기서는 위의 상대오차값을 계산하기 위해 최적해의 목적함수값 대신에 3.2절에서 제시된 한값을 사용하였다. 그 결과가 <표 3>에 나타나 있다.

## 6. 결론

본 논문에서는 최종제품조립에 필요한 부품들을 조립 전단계의 여러개의 독립된 기계를 통해 생산하는 경우에 제품조립을 효율적으로 지원할 수 있는 부품생산을 위한 일정계획 문제를 다루고 있다.

본 논문에서 제시된 부품생산일정계획은 단일기계나 두 대의 기계에 한정되었던 기존연구를 여러 대의 기계를 고려한 것으로 확장한 것이다. 하지만, 부품위주의

설계하에서 조립전단계의 일정계획을 다룬 것으로 조립 단계를 포함한 2단계 AFS를 고려한 문제로 확장될 수 있다. 또한 완료시간의 합이나 납품지연시간 등을 고려

한 보다 다양한 목적함수에 대한 연구가 진행될 수 있을 것으로 사료된다.

&lt;표 1&gt; 분지한계법의 성능평가

m	n	문제 종류	CPU 시간(sec.)		발생노드비율(%)	
			Mean	Median	Mean	Median
2	10	그룹 1	0.03	0.02	8.47E-3	3.43E-3
		그룹 2	0.01	<0.001	3.38E-3	2.04E-3
		그룹 3	0.01	<0.001	1.69E-3	4.65E-4
	12	그룹 1	0.11	0.03	2.29E-4	7.02E-5
		그룹 2	0.06	0.03	1.21E-4	7.05E-5
		그룹 3	0.03	0.02	4.88E-5	1.17E-5
	14	그룹 1	0.60	0.13	5.70E-6	1.17E-6
		그룹 2	0.56	0.12	5.26E-6	1.10E-6
		그룹 3	0.09	0.02	8.85E-7	1.29E-7
	16	그룹 1	17.41	0.68	6.03E-7	2.38E-8
		그룹 2	5.95	0.38	2.04E-7	1.23E-8
		그룹 3	1.59	0.02	5.41E-8	4.12E-10
	18	그룹 1	65.38	3.57	6.36E-9	3.53E-10
		그룹 2	9.70	0.67	9.46E-10	6.37E-11
		그룹 3	1.32	0.02	1.29E-10	4.73E-13
	20	그룹 1	1539.32	16.52	3.54E-10	4.15E-12
		그룹 2	789.52	8.85	1.76E-10	2.24E-12
		그룹 3	52.11	0.03	1.25E-11	9.04E-15
3	10	그룹 1	0.05	0.03	1.07E-2	6.81E-3
		그룹 2	0.06	0.02	1.36E-2	4.57E-3
		그룹 3	0.03	0.02	6.99E-3	1.39E-3
	12	그룹 1	0.26	0.10	3.69E-4	1.39E-4
		그룹 2	0.25	0.07	3.45E-4	8.77E-5
		그룹 3	0.04	0.02	5.36E-5	1.29E-5
	14	그룹 1	1.66	0.55	1.10E-5	3.64E-6
		그룹 2	0.83	0.22	5.50E-6	1.62E-6
		그룹 3	1.11	0.02	7.35E-6	9.61E-8
	16	그룹 1	20.91	1.35	5.01E-7	3.29E-8
		그룹 2	14.03	0.22	3.40E-7	6.85E-8
		그룹 3	1.40	0.15	3.45E-8	3.51E-9
	18	그룹 1	256.09	9.07	1.74E-8	6.19E-10
		그룹 2	58.41	3.77	3.99E-9	2.61E-10
		그룹 3	5.08	0.73	3.57E-10	5.04E-11
5	10	그룹 1	0.16	0.05	2.38E-2	7.94E-4
		그룹 2	0.12	0.05	1.55E-2	6.72E-3
		그룹 3	0.02	0.02	2.92E-3	1.73E-3
	12	그룹 1	0.66	0.33	5.98E-4	3.04E-4
		그룹 2	0.73	0.23	6.49E-4	2.17E-4
		그룹 3	0.56	0.08	4.93E-4	6.76E-5
	14	그룹 1	12.33	2.90	5.03E-5	1.23E-5
		그룹 2	3.65	1.42	1.54E-5	5.89E-6
		그룹 3	0.71	0.27	3.03E-6	1.09E-6
	16	그룹 1	89.18	8.62	1.36E-6	1.35E-7
		그룹 2	76.97	7.08	1.19E-6	1.01E-7
		그룹 3	29.26	0.28	4.64E-7	4.18E-9

&lt;표 2&gt; 발견적 방법의 성능평가(최적해와 비교)

m	n	문제 종류	HEU-1			HEU-2			HEU-3		
			Mean RE	Max. RE	Nof	Mean RE	Max. RE	Nof	Mean RE	Max. RE	Nof
2	10	그룹 1	0.030	0.091	3	0.031	0.120	1	0.004	0.018	8
		그룹 2	0.015	0.070	5	0.030	0.066	0	0.001	0.009	14
		그룹 3	0.08%	0.007	12	0.023	0.054	0	0.05%	0.006	18
	12	그룹 1	0.025	0.057	0	0.035	0.107	0	0.005	0.011	8
		그룹 2	0.014	0.032	3	0.025	0.064	0	0.005	0.033	12
		그룹 3	0.004	0.022	8	0.025	0.047	0	0.008%	0.002	19
	14	그룹 1	0.039	0.073	0	0.039	0.077	0	0.008	0.033	3
		그룹 2	0.021	0.082	2	0.023	0.057	0	0.004	0.040	9
		그룹 3	0.004	0.013	8	0.020	0.047	0	0.008%	0.002	19
	16	그룹 1	0.041	0.082	0	0.038	0.079	0	0.009	0.045	3
		그룹 2	0.027	0.072	0	0.038	0.080	0	0.003	0.024	9
		그룹 3	0.004	0.021	5	0.023	0.040	0	0.02%	0.002	17
	18	그룹 1	0.047	0.094	0	0.043	0.091	0	0.008	0.031	3
		그룹 2	0.029	0.086	0	0.039	0.075	0	0.002	0.032	7
		그룹 3	0.003	0.026	4	0.025	0.041	0	0.002%	0.06%	19
	20	그룹 1	0.045	0.108	0	0.036	0.089	0	0.011	0.035	3
		그룹 2	0.024	0.077	0	0.034	0.073	0	0.007	0.041	5
		그룹 3	0.003	0.016	7	0.032	0.069	0	0.004%	0.05%	17
3	10	그룹 1	0.032	0.006	1	0.037	0.083	0	0.009	0.042	8
		그룹 2	0.018	0.072	2	0.040	0.120	1	0.005	0.028	10
		그룹 3	0.006	0.022	6	0.029	0.146	0	0.001	0.012	15
	12	그룹 1	0.048	0.121	0	0.052	0.113	0	0.012	0.062	6
		그룹 2	0.029	0.077	0	0.037	0.073	0	0.004	0.030	11
		그룹 3	0.007	0.025	7	0.039	0.079	0	0.001	0.007	16
	14	그룹 1	0.035	0.083	0	0.039	0.093	0	0.005	0.025	5
		그룹 2	0.026	0.078	0	0.036	0.079	0	0.003	0.020	8
		그룹 3	0.006	0.025	3	0.036	0.077	0	0.04%	0.004	17
	16	그룹 1	0.047	0.090	0	0.051	0.132	0	0.009	0.029	0
		그룹 2	0.041	0.099	0	0.049	0.094	0	0.006	0.016	3
		그룹 3	0.007	0.023	3	0.042	0.089	0	0.04%	0.006	15
	18	그룹 1	0.056	0.124	0	0.045	0.098	0	0.008	0.034	0
		그룹 2	0.038	0.071	0	0.046	0.099	0	0.007	0.038	3
		그룹 3	0.009	0.029	1	0.037	0.062	0	0.02%	0.001	14
5	10	그룹 1	0.034	0.066	0	0.034	0.058	0	0.005	0.029	8
		그룹 2	0.033	0.076	0	0.055	0.138	1	0.006	0.037	4
		그룹 3	0.010	0.041	4	0.039	0.077	0	0.001	0.013	15
	12	그룹 1	0.058	0.123	0	0.049	0.116	0	0.006	0.022	1
		그룹 2	0.045	0.091	0	0.051	0.114	0	0.005	0.020	0
		그룹 3	0.010	0.033	2	0.043	0.082	0	0.02%	0.003	17
	14	그룹 1	0.034	0.08	0	0.042	0.106	0	0.005	0.030	5
		그룹 2	0.029	0.088	0	0.040	0.081	0	0.003	0.009	8
		그룹 3	0.009	0.036	2	0.044	0.115	0	0.03%	0.003	17
	16	그룹 1	0.056	0.114	0	0.057	0.103	0	0.010	0.029	0
		그룹 2	0.044	0.081	0	0.056	0.107	0	0.007	0.021	1
		그룹 3	0.008	0.037	0	0.047	0.074	0	0.08%	0.008	15

&lt;표 3&gt; HEU-3의 성능평가(하한값과 비교)

n	문제 종류	m=2		m=3		m=5	
		Mean RE	Max. RE	Mean RE	Max. RE	Mean RE	Max. RE
25	그룹 1	0.062	0.096	0.089	0.174	0.132	0.210
	그룹 2	0.023	0.079	0.040	0.149	0.068	0.202
	그룹 3	0.09%	0.014	0.003	0.033	0.005	0.028
30	그룹 1	0.065	0.104	0.091	0.131	0.131	0.208
	그룹 2	0.024	0.076	0.046	0.120	0.062	0.150
	그룹 3	0.08%	0.008	0.002	0.009	0.004	0.016
50	그룹 1	0.057	0.088	0.082	0.149	0.112	0.162
	그룹 2	0.023	0.061	0.035	0.114	0.051	0.113
	그룹 3	0.03%	0.004	0.002	0.039	0.002	0.006
80	그룹 1	0.056	0.083	0.079	0.124	0.106	0.156
	그룹 2	0.020	0.072	0.033	0.085	0.048	0.150
	그룹 3	0.02%	0.002	0.03%	0.002	0.001	0.006
100	그룹 1	0.053	0.081	0.072	0.098	0.096	0.145
	그룹 2	0.017	0.061	0.029	0.092	0.042	0.108
	그룹 3	0.01%	0.001	0.03%	0.002	0.07%	0.003

## 참고문헌

- [1] Baker, K.R., Powell, S.G., and Pyke, D.F.; "Optimal Allocation of Work in Assembly Systems," *Management Science*, 39 : 101-106, 1993.
- [2] Cheng, T.C.E., and Wang, Q.; "Scheduling the Fabrication and Assembly of Components in a Two-machine Flowshop," *IIE Transactions*, 31 : 135-148, 1999.
- [3] Coffman, E.D., Nozari, A., and Yannakakis, M.; "Optimal Scheduling of Products with Two Subassemblies on a Single Machine," *Operations Research*, 37 : 426-436, 1989.
- [4] Conway, R.W., Maxwell, W.L., and Miller, L.W.; "Theory of Scheduling," Massachusetts, Addison-Wesley, 1967.
- [5] Gerodimos, A.E., Glass C.A., and Potts, C.N.; "Scheduling the Production of Two-component Jobs on a Single Machine," *European Journal of Operational Research*, 250-259, 2000.
- [6] He, D.W., and Kusiak, A.; "Performance Analysis of Modular Products," *International Journal of Production Research*, 34 : 253-272, 1996.
- [7] Heidelberger, P. and Trivedi, K.S.; "Analytic Queueing Models for Programs with Internal Concurrency," *IEEE Transactions on Computers*, 32 : 73-98, 1983.
- [8] Lee, C.Y., Cheng, T.C.E., and Lin, B.M.T.; "Minimizing the Makespan in the 3-machine Assembly-type Flowshop Scheduling Problem," *Management Science*, 39 : 616-625, 1993.
- [9] Lin, B.M.T.; "Fabrication Scheduling on a Single Machine with Due Date Constraints," *European Journal of Operational Research*, 136 : 95-105, 2002.
- [10] Potts, C.N., Sevast'janov, S.V., Strusevich, V.A., Wassenhove, L.N., and Zwaneveld, C.M.; "The Two-stage Assembly Scheduling Problem : Complexity and Approximation," *Operations Research*, 43 : 346-355, 1995.
- [11] Sun, X., Morizawa, K., and Nagasawa H.; "Powerful Heuristics to Minimize Makespan in Fixed, 3-machine, Assembly-type Flowshop Scheduling," *European Journal of Operational Research*, 146 : 499-517, 2003.
- [12] Sung, C.S., and Park, C.K.; "Scheduling of Products with Common and Product-dependent Components Manufactured at a Single Facility," *Journal of the Operational Research Society* 44 : 773-784, 1993.
- [13] Sung, C.S., and Yoon, S.H.; "Minimizing Total Weighted Completion Time at a Pre-assembly Stage Composed of Two Feeding Machines," *International Journal of Production Economics*, 54 : 247-255, 1998.