공통 및 고유부품으로 구성되는 제품의 부품공급을 위한 단일설비 일정계획

윤 상 흠*

Fabrication Scheduling of Products with Common and Unique Components at a Single Facility

Sang-Hum Yoon*

■ Abstract ■

This paper considers a single facility scheduling problem for the fabrication of components supporting the subsequent assembly into products. Each product consists of two types of components, one is common to all products and the other is unique to itself. The unique components are processed individually and the common components are processed in batches, and a distinct setup is incurred for the production of a batch of common components. Under the so-called "batch availability" environment, we propose an efficient dynamic programming(DP) procedure for the optimal sequencing and batching schedule with minimum total completion times of products in the situation that each product is said to be complete if both common and unique components of the product are finished. Computational results show that the proposed DB procedure is more efficient than the previous work which has considered the same problem as this paper.

Keyword : Fabrication Scheduling, Batch Processing, Dynamic Programming

1. 서 론

본 연구에서는 각 제품(또는, 중간조립품)이 고유부품(unique component)과 공통부품(common component)의 두 가지 부품으로 구성되는 경우에 일정량의 제품조립을 위해 소요되는 부품들을 생

논문접수일 : 2003년 4월 1일 논문게재확정일 : 2003년 11월 3일
* 전주대학교 정보기술컴퓨터공학부 조교수

산하기 위한 단일설비 일정계획 문제를 다룬다. 고유부품이란 각 제품에 고유하게 필요한 부품을 의미하며 공통부품이란 여타 다른 제품에도 필요한 유사한 부품을 말한다. GT(group technology)의 관점에서 볼 때 공통부품들은 서로 유사한 생산적 특징을 가지므로 일회의 생산준비(setup)에 의해 여러 개의 공통부품들이 배치(batch)로 묶여 연속적으로 생산이 가능한 것들이고 고유부품들은 각 부품별로 개별적인 생산준비가 필요한 것들이다. 따라서, 공통부품의 경우에는 작업순서의 결정뿐 아니라 전체를 몇 개의 배치로, 어떤 구성으로 묶음화 할 것인가를 결정하는 배칭(batching)이 일정계획의 주요한 의사결정요소가 된다. 최종 조립의 전단계에 있는 단일설비에서 생산준비의 변환을 통해 서로 다른 형태의 부품을 생산하는 경우는 실제로 많은 제조환경에서 찾아볼 수 있으며 특별히 다양한 회로기판의 부품삽입(Coffman *et al.*[4])이나 송전변속장치의 생산(Gim and Han[7])을 예로 들 수 있다.

본 연구에서는 특별히 생산 완료된 공통부품의 이동이 개별부품단위가 아닌 생산된 배치 단위로 이루어지는 상황을 고려하고 있다. 즉, 각 공통부품의 가공이 완료되어 완제품의 조립에 활용 가능한 시점은 그 공통부품이 속한 배치의 나머지 공통부품들이 모두 생산 완료되는 시점으로 모델링된다. Santos and Magazine[10]은 이런 물류 형태를 "배치이동(batch availability)"이라고 부르고 완성된 공통부품의 이동이 생산완료시점에서 개별적으로 이루어지는 "개별이동(item availability)"과 구별하였다. 일정계획 문제에서는 일반적으로 배치이동의 경우가 개별이동의 경우보다 수리적으로 다루기 힘든 것으로 알려져 있다(Webster and Baker[16]).

각 제품의 생산을 위해서는 그 제품의 조립에 소요되는 고유부품과 공통부품이 모두 필요하므로 본 연구에서는 각 제품의 작업완료시간(job completion time)을 해당 고유부품과 공통부품배치가 모두 생산 완료되는 시점으로 모델링한다. 연구의 목적은 배치이동 상황에서 단일설비를 통해 고유부품 및 공통부품들을 생산하는 경우, 모든 작업(제품)들의 총 작업완료시간을 최소화하는 부품가공순서 및 공통부품에 대한 배칭을 결정하는 최적 일정계획해를 도출하는 것이다.

본 연구에서와 같이 단일설비를 통해 공통 및 고유부품을 가공하기 위한 작업순서와 배칭결정이 혼재된 일정계획연구는 Baker[2]를 시작으로 최근의 Lin[9]에 이르기까지 활발히 진행되어 왔으며 많은 연구결과가 보고되고 있다. 이를 공통부품배치의 이동형태와 고려된 일정계획의 목적함수에 따라 분류하면 <표 1>과 같다.

먼저 Baker[2]는 배치이동 상황에서 "어떤 작업쌍 (J_i, J_j)에 대해 작업 J_i의 공통부품 가공시간(processing time)이 J_j의 것보다 작을 경우, 두 작업의 고유부품 가공시간들에 대해서도 같은 대소관계가 항상 성립한다"는 매우 제한적인 가정 하에서 총 작업완료시간을 최소화하는 동적계획법(dynamic programming)을 제안하였다. 위의 가정을 흔히 "Baker의 가정"이라고 부른다. Aneja and Singh [1]은 Baker의 연구를 공통부품 종류가 여러개가 있는 경우로 문제를 확장하였다. Baker의 가정하에서는 최적의 작업처리순서가 고유부품 가공시간에 대한 SPT(shortest processing time)에 의해 고정되어, 결국 공통부품에 대한 배칭 결정만이 남는

〈표 1〉 공통 및 고유부품을 고려한 단일설비 일정계획 연구

	개별 이동(item availability)	배치 이동(batch availability)	
총 작업완료시간 (total completion time)	Vickson *et al.*[15] Sung and Park[13] Coffman *et al.*[4]	Baker 가정이 있는 경우	Baker[2], Aneja and Singh[1]
		Baker 가정이 없는 경우	Ding[5], Liao[8]
납기관련 함수	Lin[9]	Gerodimos *et al.*[6]	

단순화된 문제가 된다. 하지만, Baker의 가정이 없는 경우, 즉 공통 및 고유부품의 가공시간에 대한 제한이 없는 확장된 상황에서는 고유부품에 대한 작업순서와 공통부품에 대한 배칭이 상호연결된 보다 복잡한 문제가 된다. Sung and Park[13]과 Vickson *et al.*[15]은 개별이동 상황과 Baker의 가정 하에서 최적 해를 위한 분지한계법(branch and bound)과 동적계획법을 제안하였으며 Coffman *et al.*[4]은 각 작업이 두 개의 공통부품만을 필요로 하는 경우를 고려하였다. 그외의 관련연구에는 납기관련 함수(납기지연 작업의 수, 최대 지연시간 등)를 최소화하는 경우(Lin[9], Gerodimos *et al.*[6]), 단일설비가 아닌 흐름생산라인에서 최대 작업완료시간을 최소화하는 경우(Cheng and Wang[3], Sung *et al.*[12]), 병렬설비를 통해 공통부품과 고유부품을 각기 따로 생산하는 경우(Sung and Yoon[14]) 등이 있다.

본 연구는 배치이동을 고려하고 Baker의 가정이 없는 확장된 상황에서 총 작업완료시간을 최소화하는 일정계획문제를 다루고 있으며 최적해의 도출을 위한 효율적인 동적계획법을 제안한다. 본 연구와 동일한 문제를 다룬 기존연구에는 Ding[5]과 Liao[8]의 결과가 있다. Ding[5]은 반복적 작업교환(job interchange)을 이용한 몇가지 발견적 해법을 제시하였다. 하지만, 제안된 발견적 해의 성능평가를 위한 최적해의 도출을 위해서는 가능한 모든 해를 나열하는 방법을 사용하여 작업수가 8개 미만인 경우로 매우 제한된 실험결과만을 제시하였다. Liao[8]는 문제에 대한 몇가지 성질분석을 통해 최적 일정계획해를 위한 동적계획법을 최초로 제안하였다. 본 연구에서는 연속된 생산준비사이에 가공되는 공통 및 고유부품들을 배치의 확장개념인 런(run)으로 묶음화 함으로써 문제를 부품단위가 아닌 작업(제품)단위의 해법으로 접근할 수 있으며 이를 활용하면 전체문제가 우선순위관계(precedence relation)를 가지는 단일설비 일정계획문제로 단순화 될 수 있음을 보인다. 또한 이러한 결과에 바탕하여 새롭게 제안된 동적계획법이 Liao[8]의 결과보다 매우 효율적임을 수치실험을 통해 증명한다. 본 논문의 구성은 다음과 같다. 2장에서는 문제의 정확한 이해를 돕기 위해 고려하고 있는 일정계획문제에 대한 수리적 정의를 하고 3장에서는 문제에 대한 몇 가지 성질규명과 이를 활용한 새로운 동적계획법이 제안된다. 4장에서는 수치실험을 통해 제안된 동적계획법이 기존연구에 비해 매우 효율적임을 증명한다. 마지막으로 5장에서는 결론과 추후연구방향에 대한 논의가 이루어 진다.

2. 문제의 구성

대상 문제와 분석에 사용될 기호를 정의하면 다음과 같다. 단일설비에서 수행해야 할 총 n개의 작업이 존재하고, 각 작업 $J_i\,(i=1,2,\cdots,n)$에 해당하는 공통부품과 고유부품의 가공을 위해서는 각각 c_i, u_i의 가공시간이 필요하다. 공통부품은 배치단위로 묶여서 처리되며 각 배치의 처리를 위해서는 준비시간 $\triangle$가 필요하지만, 고유부품의 경우는 개별적인 생산준비를 가지므로 준비시간이 가공시간 u_i에 포함되어 있다. 공통부품배치의 크기에 대한 제한은 없으며 완료된 공통부품의 이동은 배치단위로 이루어지는 배치이동을 가정한다. 본 논문에서는 설명의 편의를 위해 혼동되지 않는 범위내에서 기호 c_i, u_i를 각 부품에 대한 가공시간 뿐 아니라 특정 부품 자체를 표시하기 위해서도 사용된다. 각 작업의 작업완료시간은 해당하는 고유부품과 공통부품이 모두 생산 완료되어 활용가능한 시점이 된다. 예를 들어, 4개의 작업(8개의 부품)에 대한 임의의 일정계획해 $S_1=u_1u_2\triangle c_1c_2c_3u_3u_4\triangle c_4$를 고려할 때([그림 1] 참조), 각 작업 J_i의 작업완료시간 f_i를 계산하면,

$$f_1=f_2=u_1+u_2+\triangle+c_1+c_2+c_3,$$

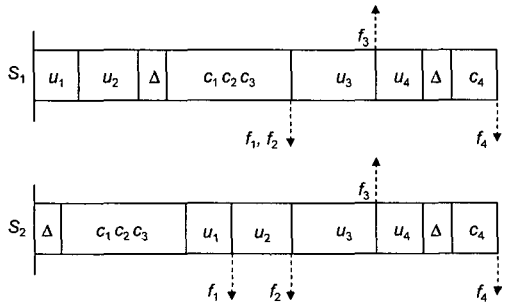
$$f_3=f_1+u_3,$$

$$f_4=2\triangle+\sum_{i=1}^{4}(c_i+u_i)$$

와 같다. 동일한 문제에 대해 S_1과 유사하지만 c_1과 c_2가 속한 공통부품배치와 고유부품간의 가공순서를 서로 바꾼 새로운 일정계획해 $S_2 = \triangle c_1 c_2 c_3 u_1 u_2 u_3 u_4 \triangle c_4$에 대한 작업완료시간을 계산하면 f_2, f_3, f_4는 S_1의 경우와 동일하나 J_1의 경우에는

$$f_1 = u_1 + \triangle + c_1 + c_2 + c_3$$

로써 S_1의 경우보다 완료시간이 u_2만큼 줄어들게 됨을 알 수 있다. 이 예제를 통해 작업완료시간은 공통부품에 대한 배칭뿐 아니라 부품간의 처리순서에 따라 복합적으로 결정됨을 알 수 있다. 본 연구의 목적은 이러한 작업완료시간들을 합한 총 작업완료시간 $\sum_{i=1}^{n} f_i$를 최소화하는 일정계획해 S^*(전체 부품의 가공순서 및 공통부품에 대한 배칭)를 구하는 것이다.



[그림 1] 두 개의 일정계획해에 대한 간트차트

3. 새로운 동적계획법

본 연구에서 제안하는 동적계획법은 다음의 네 가지 성질들을 효과적으로 활용한 것이다. [성질 1]과 [성질 2]는 관련문헌([2, 5, 8])에서 밝혀진 것들로 증명은 생략한다.

[성질 1] 모든 작업 J_i에 대해 항상 공통부품(c_i)이 고유부품(u_i)보다 앞서서 가공되는 최적의 일정계획해가 존재한다.

[성질 2] 임의의 작업 쌍 (J_i, J_j)에 대해 다음의 조건이 만족되면,

$$c_i \le c_j \text{ 이고 } u_i \le u_j, \qquad (1)$$

c_i는 c_j보다 앞서서 가공되고, u_i는 u_j보다 앞서서 가공되는 최적의 일정계획해가 존재한다.

[성질 1]은 한 작업의 공통 및 고유부품간에 우선순위관계가 존재함을 나타내고, [성질 2]는 조건 (1)을 만족하는 작업들의 부품 쌍간에는 SPT의 우선순위관계가 존재함을 보여준다. 이를 활용하면 [그림 2(a)]에서 예시되는 것과 같이 부품간의 우선순위관계를 순위관계 그래프로 표현할 수 있다. 참고로 Liao[8]가 제안한 동적계획법에서는 이러한 부품간 순위관계를 어기지 않으면서 전체부품집합에 도달할 때까지 부분적인 부품집합(partial component set)을 점차적으로 확장하는 방법을 사용하였다.

다음에 나오는 성질들은 총 $2n$개의 부품을 각각 취급할 필요가 없이 n개의 작업단위로 문제를 접근할 수 있음을 보여준다. 증명의 편의를 위해 두 개의 연속된 생산준비들 사이에 위치한 부분부품가공순서(partial component sequence)를 런(run)이라고 정의하고 런간의 순서를 고려하여 i 번째 런을 r_i로 표시한다. 예를 들어, 어떤 일정계획해 $S = \triangle c_1 c_2 u_1 \triangle c_3 u_2 u_3 \triangle c_4 u_4$의 경우, 3개의 런($r_1 = c_1 c_2 u_1$, $r_2 = c_3 u_2 u_3$, $r_3 = c_4 u_4$)이 존재하게 된다.

[성질 3] 최적 일정계획해에서 특정 공통부품 c_i가 어떤 런에 속해 있을때, 그 공통부품에 대응하는 고유부품 u_i도 같은 런에 속하는 최적해가 존재한다.

증명. 만약 어떤 최적일정계획해에서 c_i와 u_i가 서로 다른 런에 배치되어 있다고 가정할 때, 다음의 두 가지 경우가 발생하며 각각의 경우에 적절한 부품이동을 통해 총 작업완료시간을 증가시키지

않으면서 두 부품을 같은 런에 배치시킬 수 있다.

경우 1 : c_i가 u_i보다 순서상 앞선 경우,

c_i를 현재의 위치에서 빼내어 u_i가 속한 런에 포함된 공통부품배치에 삽입한다. 배치이동 상황에서는 공통부품배치안에서의 구체적인 삽입위치는 고려될 필요가 없다.

경우 2 : u_i가 c_i보다 앞선 경우,

u_i를 현재의 위치에서 빼내어 c_i가 속한 공통부품배치의 바로 다음 처리 위치에 삽입한다.

[성질 4] 최적 일정계획해에서 임의의 런에 속한 고유작업들간의 가공순서는 u_i 값들에 대한 SPT 순서를 만족한다.

증명. 배치이동 상황과 단일설비 일정계획문제에서 SPT순서가 총 작업완료시간을 최소화함을 상기할 때 이 결과는 당연한 것이다.

위의 [성질 3]과 [성질 4]에 의해 어떤 작업의 공통 및 고유부품은 같은 런에 포함되어야 하고 런내부에서의 부품처리순서는 u_i 값의 SPT 순서로 고정되어야 함을 알 수 있다. 예를 들어, 작업들의 표시(index)가 u_i 값의 오름차순으로 정렬되어 있다고 할 때, $S = \triangle c_3 c_2 u_3 \triangle c_1 u_2 u_1$의 경우 위의 성질들을 활용하면 $S' = \triangle c_3 c_2 u_2 u_3 \triangle c_1 u_1$나 $S'' = \triangle c_3 u_3 \triangle c_1 c_2 u_1 u_2$로 바뀌는 것이 유리하다. 더욱이 일정계획해에 대한 표현이 부품단위가 아닌 작업단위로 가능해지므로, 일정계획해의 표현도 $S' = \triangle J_2 J_3 \triangle J_1$, $S'' = \triangle J_3 \triangle J_1 J_2$와 같이 간략화될 수 있다. 이와 같이 작업단위로 문제를 접근하는 것은 [성질 2]를 함께 활용할 경우 [그림 2(b)]에서 예시되는 바와 같이 작업들간 우선순위관계가 있는 상황에서 n개의 작업들에 대한 단일설비 일정계획문제로 단순화될 수 있는 가능성을 보여준다.

지금부터, 위의 4가지 성질들을 모두 활용한 새로운 동적계획법을 설명한다. 먼저, 작업들의 표시는 해당 u_i 값들의 오름차순으로 정렬되어 있으며, [성질 2]를 활용하여 작업들간 우선순위관계를 모두 구했다고 가정한다.

동적계획법의 설명에 사용될 기호들을 정의하면 다음과 같다.

- Ω : 모든 작업들의 집합, $\Omega = \{J_1, J_2, \cdots, J_n\}$,
- B : 작업들의 부분집합, $B \in \Omega$,
- L_B : B에 속한 작업들로만 구성된 부분 순위관계그래프에서 후속작업이 없는 작업들의 집합, $L_B \in B$,
- R_B : B를 처리할 때, 작업간 우선순위관계를 어기지 않으면서 처리순서상 마지막에 위치할 수 있는 가능한 모든 런들의 집합,
- SC_r : 시간 0에서 시작하여 런 r을 처리할 때 r에 속한 작업들의 완료시간의 합(sum of completion time),
- MC_r : 시간 0에서 시작하여 런 r을 처리할 때 r의 작업완료시간,
- $F_r(B)$: $r \in R_B$인 런 r이 마지막에 위치한다는 조건하에서, 시간 0에서 시작하여 B에 속한 작업들을 모두 처리하면서 발생되는 총 작업완료시간($\sum_{i=1}^{n} f_i$)의 값에 기여하는 최소의 증분치(minimal contribution),
- $G(B)$: 시간 0에서 시작하여 B에 속한 작업들을 처리하면서 발생되는 총 작업완료시간의 값에 기여하는 최소의 증분치.

위의 기호들을 이용하여 동적계획에 필요한 반복식(recursive equation)을 구성하면 다음과 같다.

$$G(B) = \min_{r \in R_B} \{F_r(B)\}, \quad (2)$$

$$F_r(B) = G(B \backslash r) + SC_r + (n - |B|) \cdot MC_r. \quad (3)$$

여기서 $B\backslash r$은 B에서 런 r에 속한 작업들을 제외한 나머지 집합을 나타내며, $|B|$는 B에 속한 작업들의 수를 나타낸다. 동적계획의 진행은 초기값 $G(B=\varnothing)=0$에서 출발하여 B를 점차 확장하다가 $G(B=\Omega)$에서 종료된다. Schrage and Baker [11]는 우선순위관계를 가지는 작업들에 대한 단일설비일정계획을 위한 동적계획법을 제안하면서 B와 같은 부분작업집합을 우선순위관계를 어기지 않으면서 Ω에 도달할 때까지 확장하는 효과적인 나열절차(enumerative procedure)를 제시한 바가 있다. 본 연구에서 제안하는 동적계획법에서도 B의 확장순서는 Schrage and Baker[11]의 나열절차를 사용한다. 나열절차에 대한 자세한 내용은 여기서는 생략한다.

식 (2)와 식 (3)에서 R_B에 속한 런의 나열과 SC_r, MC_r의 계산은 B 이전의 정복된 상태(state)에서 얻어진 결과를 참조하면 매우 효율적일 수 있다. 만약 현재 동적계획의 탐색상태가 B에 도달했으며 L_B를 구했다고 가정하자. L_B에 속한 작업들은 B를 처리할때 마지막 런의 마지막 위치에 포함되어야 하는 작업들을 의미한다. 따라서, $J_i \in L_B$인 작업 J_i는 단독으로 런을 구성할 수도 있으며, B에서 J_i를 제외한 상태 $B\backslash J_i$에 해당하는 $R_{B\backslash J_i}$에 속한 각각의 런 $r' \in R_{B\backslash J_i}$에 대해 r'의 마지막에 J_i를 삽입한 $r = r'J_i$를 모두 나열하면 상태 B에서 가능한 마지막 런들의 집합 R_B를 쉽게 구할 수 있다. 이를 수식으로 표현하면 $R_B = \{J_i, r'J_i \mid J_i \in L_B, r' \in R_{B\backslash J_i}\}$가 된다. 이때, 어떤 r'의 마지막 작업이 J_k라고 할 때, 만약 $k > i$이면 [성질 4]에 의해 $r'J_i$는 고려 대상에서 제외된다. 또한 MC_r과 SC_r의 계산도 상태 $B\backslash J_i$에서 계산된 $SC_{r'}$과 $MC_{r'}$로 부터

$$MC_r = \begin{cases} \triangle + c_i + u_i & |r| = 1 \text{일때 } (r = J_i \text{라면}) \\ MC_{r'} + c_i + u_i & |r| > 1 \text{일때} \end{cases} \quad (4)$$

$$SC_r = \begin{cases} \triangle + c_i + u_i & |r| = 1 \text{일때 } (r = J_i \text{라면}) \\ SC_{r'} + |r'| \cdot c_i + MC_r & |r| > 1 \text{일때} \end{cases} \quad (5)$$

를 활용하면 쉽게 계산된다.

제시된 동적계획법에 대한 이해를 돕기 위한 예제를 설명한다. 여기서 사용되는 예제는 효율성 비교를 위해 Liao[8]에서 사용한 예제와 동일한 것을 채택하였다. 예제는 5개의 작업을 가지고 있으며, 각 부품에 대한 가공시간은 $c_i = \{1, 4, 2, 5, 3\}$, $u_i = \{2, 3, 4, 6, 8\}$이다. 공통부품배치를 위한 준비시간은 $\triangle = 2$이다.

[성질 2]에 의해 구해진 작업들간 순위관계 그래프가 [그림 2(b)]와 같이 나타난다. 따라서 동적계획에 사용될 B의 진행(Schrage and Baker[11]의 나열절차를 참조)은 $\{J_1\}$, $\{J_1, J_2\}$, $\{J_1, J_3\}$, $\{J_1, J_2, J_3\}$, $\{J_1, J_2, J_3, J_4\}$, $\{J_1, J_3, J_5\}$, $\{J_1, J_2, J_3, J_5\}$, $\{J_1, J_2, J_3, J_4, J_5\}$의 순서로 전개된다. 이에 따른 동적계획법의 전체 계산 결과가 <표 2>에 나타나 있다. 만약, 현재 상태가 $B = \{J_1, J_3, J_5\}$에 있다고 할때, [그림 2(b)]로 부터 $L_B = \{J_5\}$임을 알 수 있다. 따라서, B를 처리할 때 J_5는 마지막 런에 포함되어야 하며, 가능한 런들의 집합 R_B는 J_5를 단독으로 런을 구성하는 것($r_1 = J_5$)과 B 이전에 탐색된 상태 $B\backslash J_5 = \{J_1, J_3\}$에서 얻어진 $R_{B\backslash J_5} = \{J_3, J_1, J_3\}$의 각 요소에 J_5를 삽입하는 경우를 포함하여 최종적으로 $R_B = \{r_1 = J_5, r_2 = J_3J_5, r_3 = J_1J_3J_5\}$임을 알 수 있다. 여기서 식 (4)와 식 (5)를 활용하면,

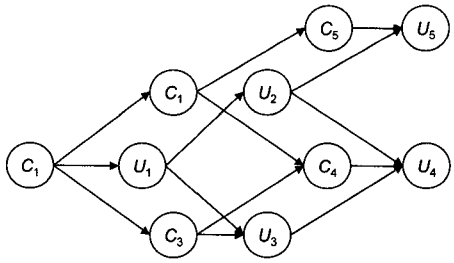
$$\begin{aligned} MC_{r_1} &= SC_{r_1} = \triangle + c_5 + u_5 = 2 + 3 + 8 = 13, \\ MC_{r_2} &= MC_{J_3} + c_5 + u_5 = 8 + 3 + 8 = 19, \\ SC_{r_2} &= SC_{J_3} + |J_3| \cdot c_5 + MC_{r_2} = 8 + 3 + 19 = 30, \\ MC_{r_3} &= MC_{J_1J_3} + c_5 + u_5 = 11 + 3 + 8 = 22, \\ SC_{r_3} &= SC_{J_1J_3} + |J_1J_3| \cdot c_5 + MC_{r_3} = 18 + 2 \times 3 \\ &\quad + 22 = 46 \end{aligned}$$

를 계산할 수 있다. 또한, 식 (2)와 식 (3)을 활용하면

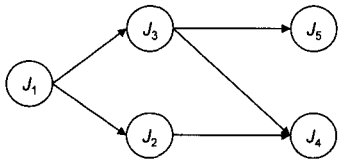
$$G(B) = \min_{r_i \in R_B}\{G(B\backslash r_i) + SC_{r_i} + (n - |B|) \cdot MC_{r_i}\}$$
$$= \min\{51 + 13 + 2\times 13,\ 25 + 30 + 2\times 19,\ 0 + 46 + 2\times 22\}$$
$$= 90$$

임을 알 수 있다. $G(B)$의 최소값 선택에서 동일한 값들이 있는 경우에는 임의의 것을 선택한다. 이와 같이 동적계획을 계속 진행하면 주어진 예제에 대한 최적해는 <표 2>에서 계산된 바와 같이 $S^* = \triangle J_1 J_3 \triangle J_2 J_5 \triangle J_4 = \triangle c_1 c_3 u_1 u_3 \triangle c_2 c_5 u_2 u_5 \triangle c_4 u_4$로 얻어지며 총 작업완료시간은 116이다.

주어진 예제의 경우 <표 2>에서 보는 바와 같이 총 8개의 상태가 탐색되었으며, 참고로 Liao[8]의 경우에는 같은 예제에 대해 총 29개의 상태가 탐색되었다.



(a) Liao[8]에 의한 부품간 순위관계 그래프



(b) 제안된 동적계획법을 위한 작업간 순위관계 그래프

[그림 2] 순위관계 그래프의 예시

〈표 2〉 예제에 대한 풀이

B	L_B	R_B	MC_r	SC_r	$F_r(B)$	$G(B)$
$\{J_1\}$	$\{J_1\}$	J_1	5	5	25	25
$\{J_1, J_2\}$	$\{J_2\}$	J_2 J_1J_2	9 12	9 21	61 57	57
$\{J_1, J_3\}$	$\{J_3\}$	J_3 J_1J_3	8 11	8 18	57 51	51
$\{J_1, J_2, J_3\}$	$\{J_2, J_3\}$	J_2, J_3 J_2J_3 $J_1J_2J_3$	9, 8 15 18	9, 8 26 43	78, 81 81 79	78
$\{J_1, J_2, J_3, J_4\}$	$\{J_4\}$	J_4 J_2J_4, J_3J_4 $J_2J_3J_4$ $J_1J_2J_3J_4$	13 20, 19 26 29	13 34, 32 62 87	104 105, 108 113 116	104
$\{J_1, J_3, J_5\}$	$\{J_5\}$	J_5 J_3J_5 $J_1J_3J_5$	13 19 22	13 30 46	90 93 90	90
$\{J_1, J_2, J_3, J_5\}$	$\{J_2, J_5\}$	J_2, J_5 J_2J_5, J_3J_5 $J_2J_3J_5$ $J_1J_2J_3J_5$	9, 13 20, 19 26 29	9, 13 32, 30 58 81	108, 104 103, 106 109 110	103
$\{J_1, J_2, J_3, J_4, J_5\}$	$\{J_4, J_5\}$	J_4, J_5 J_2J_4, J_4J_5 $J_2J_4J_5$, $J_3J_4J_5$ $J_2J_3J_4J_5$ $J_1J_2J_3J_4J_5$	13, 13 20, 24 31, 30 37 40	13, 13 34, 40 71, 68 108 139	116, 117 124, 118 122, 125 133 139	116

4. 실험결과

본 절에서는 제시된 동적계획법에 대한 성능평가를 위해 수치 데이터를 통한 실험결과를 보여준다. 실험은 Pentium PC/997MHz에서 수행되었으며 알고리즘의 프로그래밍을 위해서는 C언어를 사용하였다. u_i의 값은 동일하게 [1,99]의 범위를 가지는 일양분포로부터 무작위로 추출되었으며, c_i의 값은 $[\alpha \cdot u_i, (1-\alpha) \cdot u_i]$의 범위를 가지는 일양분포로부터 생성되었는데 문제종류 I의 경우는 $\alpha = 0.2$, 문제종류 II는 $\alpha = 0.1$, 문제종류 III은 $\alpha = 0$의 값을 사용하였다. α 값이 클수록 c_i 값의 생성범위가 좁아지므로 [성질 2]로 부터 얻어지는 작업들간의 우선순위관계가 많아지고, 따라서 보다 쉬운 문제가 생성된다. 작업수는 15개 부터 50개에 이르기 까지 다양한 경우를 고려하였다. 각 작업수와 문제종류의 조합에 대해 50개씩의 문제를 생성하여 관련 통계량을 추출하였다. 성능비교는 각 동적계획법이 탐색한 상태수와 사용된 CPU시간을 기준으로 하였다. 실험의 결과는 <표 3>에 요약되어 있다. 표에서 보는 바와 같이 제안된 동적계획법이 Liao의 경우보다 매우 우수하다. 특히 작업수의 증가에 따른 탐색상태의 수의 증가율이 제안된 동적계획법의 경우 크지 않는 장점을 가진다.

〈표 3〉 수치실험 결과

n	문제종류	Liao[8]의 동적계획법				제안된 동적계획법			
		탐색상태수		CPU 시간(sec)		탐색상태수		CPU 시간	
		평 균	최 대	평 균	최 대	평 균	최 대	평 균	최 대
15	I	581.1	1,395	0.006	0.06	35.2	63	< 0.001	< 0.001
	II	975.0	2,717	0.014	0.06	47.6	95	< 0.001	< 0.001
	III	1,343.5	6,705	0.020	0.22	58.0	185	0.001	0.05
20	I	1,682.4	6,796	0.043	0.27	60.8	149	0.001	0.05
	II	2,815.6	11,253	0.076	0.49	83.4	205	0.002	0.06
	III	4,936.8	22,353	0.172	1.48	113.3	304	0.002	0.06
25	I	4,058.8	12,656	0.132	0.55	98.5	199	0.003	0.06
	II	7,926.4	33,263	0.366	3.07	142.2	387	0.009	0.06
	III	14,647.5	59,281	1.095	10.28	200.9	620	0.010	0.11
30	I	10,587.4	36,558	0.595	3.24	159.6	370	0.004	0.06
	II	22,154.2	140,542	2.701	49.30	243.9	904	0.012	0.14
	III	41,660.9	161,869	8.188	79.10	340.5	911	0.015	0.17
35	I	22,695.6	69,587	2.092	11.59	239.6	565	0.012	0.06
	II	46,810.0	232,839	10.11	167.7	365.2	1,009	0.014	0.16
	III	112,584.4	606,336	94.16	1,969.6	568.1	1,569	0.044	0.27
40	I	5,254.2	391,639	20.43	487.6	355.1	1,343	0.017	0.17
	II	107,061.3	387,818	51.45	632.7	554.4	1,238	0.035	0.17
	III	‡	‡	‡	‡	906.3	2,780	8.712	132.9
50	I	187,651.3	1,003,282	220.2	2,564.4	710.0	1,988	0.439	8.24
	II	‡	‡	‡	‡	1,268.7	4,759	10.95	143.4
	III	‡	‡	‡	‡	2,129.3	9,123	88.50	635.3

주) ‡ : 주어진 CPU 시간(3,600sec)이나 메모리용량을 초과하는 문제가 발생한 경우임.

5. 결론 및 추후연구

본 연구에서는 고유 및 공통부품으로 구성되는 제품의 조립을 위해 소요되는 부품의 생산을 위한 단일설비 일정계획 문제를 다루었다. 총 작업완료시간의 최소화를 목적으로 하는 경우를 대상으로 활용이 쉽고 보다 효율적인 새로운 동적계획법이 제시되었다. 본 연구에서는 공통부품에 대한 배치이동과 Baker의 가정이 없는 상황을 고려함으로써 기존의 유사연구에 비해 가장 복잡성이 높고 일반화된 문제를 대상으로 하였다.

본 연구는 제품설계단계에서 주조립공정을 단순화하기 위한 부품위주설계(fabrication-driven design) 또는 모듈러설계(modular design)를 채택한 경우를 내재하고 있으며 이는 부품생산단계가 조립단계에 비해 상대적으로 병목(bottleneck)공정이 됨을 의미하게 된다. 이를 일반화하여 생산된 부품을 완성품으로 조립하는 주조립단계까지를 포함하는 2단계 흐름조립라인을 대상으로 하는 일정계획 문제나 부품생산단계에서도 단일설비가 아닌 병렬설비를 고려한 부품생산일정계획문제 등은 추후 연구과제로 고려될 수 있다.

참 고 문 헌

[1] Aneja, Y.P. and N. Singh, "Scheduling Production of Common Components at a Single Facility," *IIE Transactions*, Vol.22(1990), pp.234-237.

[2] Baker, K.R., "Scheduling the Production of Components at a Common Facility," *IIE Transactions*, Vol.20(1988), pp.32-35.

[3] Cheng, T.C.E. and Q. Wang, "Scheduling the Fabrication and Assembly of Components in a Two-machine Flowshop," *IIE Transactions*, Vol.31(1999), pp.135-148.

[4] Coffman, E.D., A. Nozari and M. Yannakakis, "Optimal Scheduling of Products with Two Subassemblies on a Single Machine," *Operations Research*, Vol.37(1989), pp.426-436.

[5] Ding, F.Y., "A Pairwise Interchange Solution Procedure for a Scheduling Problem with Production of Components at a Single Facility," *Computers and I.E.*, Vol.18(1990), pp.325-331.

[6] Gerodimos, A.E., C.A. Glass and C.N. Potts, "Scheduling the Production of Two-component Jobs on a Single Machine," *European Journal of Operational Research*, Vol. 20(2000), pp.250-259.

[7] Gim, B. and M.H. Han, "Economic Scheduling of Products with *N* Components on a Single Machine," *European Journal of Operational Research*, Vol.96(1997), pp. 70-577.

[8] Liao, C.J., "Optimal Scheduling of Products with Common and Unique Components," *International Journal of Systems Science*, Vol.27(1996), pp.361-366.

[9] Lin, B.M.T, "Fabrication Scheduling on a Single Machine with Due Date Constraints," *European Journal of Operational Research*, Vol.136(2002), pp.95-105.

[10] Santos, C.A. and M.J. Magazine, "Batching in Single Operation Manufacturing Systems," *Operations Research Letters*, Vol.4 (1985), pp.99-103.

[11] Schrage, L. and K.R. Baker, "Dynamic Programming Solution of Sequencing Problems with Precedence Constraints," *Operations Research*, Vol.26(1978), pp.444-449.

[12] Sung, C.S., Y.H. Kim and S.H. Yoon, "A Problem Reduction and Decomposition Approach for Scheduling for a Flowshop of Batch Processing Machines," *European*

Journal of Operational Research, Vol.121 (2000), pp.179-192.

[13] Sung, C.S. and C.K. Park, "Scheduling of Products with Common and Product-dependent Components Manufactured at a Single Facility," *Journal of the Operational Research Society*, Vol.44(1993), pp.773-784.

[14] Sung. C.S. and S.H. Yoon, "Minimizing Total Weighted Completion Time at a Preassembly Stage Composed of Two Feeding Machines," *International Journal of Production Economics*, Vol.54(1998), pp.247-255.

[15] Vickson, R.G., M.J. Magazine and C.A. Santos, "Batching and Sequencing of Components at a Single Facility," *IIE Transactions*, Vol.25(1993), pp.65-70.

[16] Webster, S. and K.R. Baker, "Scheduling Groups of Jobs on a Single Machine," *Operations Research*, Vol.43(1995), pp.692-703.