

두 단계 조립시스템에서 총 가중완료시간을 최소화하는 일정계획문제

윤상흠^{1*} · 이익선² · 이종협³

¹영남대학교 경영학부 / ²한국과학기술원 산업공학과 / ³인제대학교 정보통신공학과

A Scheduling Problem to Minimize Weighted Completion Time in the Two-stage Assembly-type Flowshop

Sang Hum Yoon¹ · Ik Sun Lee² · Jong Hyup Lee³

¹School of Management, Yeungnam University, Gyeongsangbuk-do 712-749

²Department of Industrial Engineering, KAIST, Daejeon 305-701

³Department of Information and Communications Engineering, Inje University, Gyeongsangnam-do 621-749

This paper considers a scheduling problem to minimize the total weighted completion time in the two-stage assembly-type flowshop. The system is composed of multiple fabrication machines in the first stage and a final-assembly machine in the second stage. Each job consists of multiple components, each component is machined on the fabrication machine specified in advance. The manufactured components of each job are subsequently assembled into a final product on the final-assembly machine. The objective of this paper is to find the optimal schedule minimizing the total weighted completion time of jobs. Three lower bounds are derived and tested in a branch-and-bound (B&B) procedure. Also, three heuristic algorithms are developed based on the greedy strategies. Computational results show that the proposed B&B procedure is more efficient than the previous work which has considered the same problem as this paper.

Keywords: Scheduling, Assembly System, Branch-and-Bound

1. 서론

주문조립방식(make-to-assembly)의 생산시스템에서는 각 부품 또는 구성품이 서로 다른 설비와 경로를 통해 독립적으로 생산된 후 최종 조립공정을 거쳐 제품으로 완성되는 2단계 구조가 일반적이다. 이 경우 개별 부품생산설비와 조립설비 간에는 물리적으로 2단계 직렬구조를 이루지만 부품 생산설비들 간에는 독립적인 병렬구조를 가짐으로써 전체적으로는 직렬과 병렬이 혼재된 형태를 이루게 된다. 직렬형 생산라인을 따라서 반제품이 이동하면서 작은 부품들이 조립됨으로써 점차 완제품화 되는 흐름라인(flowshop)의 경우와 달리 2단계 조립

시스템에서는 개별 부품들이 병렬전용기계를 통해 개별적으로 직접 생산되거나, 외부 협력업체를 통해 조달된 후 최종 조립을 위해 마련된 일정한 장소에서 완제품으로 최종 조립된다. 이러한 형태는 흐름라인에 비해 제품이나 부품의 부피가 매우 크고, 주문형 생산방식의 경우에 적합하며 고정조립시스템(fixed assembly system)으로 불리기도 한다(He and Kusiak, 1996).

본 연구는 2단계 조립시스템에서의 일정계획 문제를 다루고 있다. 시스템의 전 단계는 여러 대의 부품생산을 위한 병렬설비로 구성되고, 후 단계는 최종 조립을 위한 설비로 이루어진다. 각 제품의 최종 조립은 해당 제품의 모든 부품이 생산되

* 연락저자 : 윤상흠, 712-749 경북 경산시 대동 214-1번지 영남대학교 경영학부, Tel : 053-810-2728, Fax : 053-810-4652,
E-mail : shyoon@yumail.ac.kr

2006년 06월 접수; 2006년 09월 수정본 접수; 2006년 11월 게재 확정.

어 준비된 이후 시작될 수 있으며, 각 제품의 최종 완료시간은 최종 조립이 완료된 시점으로 모델링된다. 본 연구는 이러한 구조에서 총 가중완료시간(total weighted completion time)을 최소화 할 수 있는 각 설비에서의 작업순서를 찾는 일정계획 문제를 다루고 있다. 일정계획에서 총 가중완료시간은 가중치를 고려한 각 제품의 가중완료시간을 전체작업에 대해 합한 것을 의미하며, 모든 제품을 생산 완료하는 최종시점을 의미하는 최종 작업완료시간(makespan)이나 가중치를 고려하지 않는 단순완료시간의 합(sum of completion time)에 비해 문제의 복잡도가 가장 높은 성과척도이다. 총 가중완료시간을 일정계획의 성과척도로 사용하면 각 제품별 중요도(가중치)를 감안하여, 단위 생산시간당 중요도가 높은 제품일수록, 다른 제품들에 우선해서 재공재고량을 최소화하게 된다(Baker, 1974).

본 연구에서 고려하고 있는 일정계획문제는 NP-complete에 속한 문제로써, 작업(제품)수나 전 단계에 속한 부품설비의 수가 클 경우에는 최적해를 구하기가 힘들다 따라서, 본 연구에서는 크기가 작은 문제(대략 30작업 이하)에서 최적해를 도출할 수 있는 분지한계법(branch-and-bound, 이후 B&B)과 크기가 큰 문제에 대해서도 빠른 시간에 효과적인 해를 제공할 수 있는 휴리스틱 알고리즘을 함께 제안하게 된다.

2단계 조립시스템에 대한 연구는 먼저 확률적(stochastic) 대기행렬이론에서 컴퓨터병렬처리를 위한 시스템분석에 활용되어 왔다. 대기행렬에서는 이러한 구조를 유사조립대기행렬(assembly-like queue)이나 포크/조인(fork/join)으로 부르기도 한다(Bhat, 1986; Duda and Czachorski, 1987; Norahari and Sundarajan, 1995). Baker and Powell(1993)과 Baker *et al.*(1993)은 시뮬레이션과 확률적 분석을 통해 2단계 조립시스템의 생산률(throughput)을 최대화하기 위한 작업부하의 할당에 대해 연구하였다.

본 연구에서와 같이 확정적(deterministic) 일정계획분야에서 2단계 조립시스템을 다룬 연구는 Lee *et al.*(1993)로 부터 시작된다. Lee *et al.*은 전 단계에 2대의 부품생산설비와 후 단계에 한 대의 조립설비로 구성되는 2단계 시스템에서 최종 작업 완료시간을 최소화하기 위한 일정계획문제를 다루었다 대상 문제가 NP-complete임을 증명하였고, 부품생산설비들을 가상적인 단일기계로 집산화(aggregation)함으로써 전체문제를 2단계 흐름라인으로 근사화(approximation)하고 Johnson's rule을 활용한 3가지 휴리스틱을 제안하였다. Sun *et al.*(2003)은 동일한 문제에 대한 14가지의 새로운 휴리스틱을 제안하였다. Lee *et al.*의 휴리스틱은 이론적인 오차한계(error bound)를 도출함으로써 성능을 검증한 반면 Sun *et al.*의 휴리스틱은 난수 데이터를 활용한 수치실험으로 성능을 비교하였다. Potts *et al.*(1995)은 Lee *et al.*의 연구를 전단계의 부품생산설비가 2대 이상인 경우로 확장한 연구를 수행하였다. Hariri and Potts(1997)는 동일한 문제에 대해 최적해를 도출할 수 있는 B&B

알고리즘을 제안한 바 있다. 이상으로 소개된 연구들은 모두 목적함수가 최종 작업완료시간인 경우를 고려하였으며 총 가중완료시간을 목적함수로 가지는 본 연구와는 차별화 된다.

Tozkapan *et al.*(1993)은 본 연구와 동일하게 총 가중완료시간을 최소화 하는 문제를 제안하고 최적해의 도출을 위한 B&B 알고리즘과 B&B의 상한값으로 활용할 수 있는 휴리스틱을 제시한 바 있다. 본 연구에서는 문제에 대한 새로운 정수 계획모형을 제안하고, 개별 제약식들을 완화(relaxation)하는 기법을 활용하여 Tozkapan *et al.*에 비해 보다 우수한 새로운 3가지 하한값(lower bounds)을 제안한다. 또한, 그리디(greedy)방식을 활용한 3가지 휴리스틱 알고리즘도 개발하였다. 이러한 하한값과 휴리스틱을 B&B 알고리즘에 활용할 경우 계산시간이나 발생노드 등의 B&B의 성능이 전반적으로 향상됨을 수치실험을 통해 검증한다.

본 연구에서는 표현의 편의를 위해 Lee *et al.*(1993)에서와 같이 2단계 조립시스템을 AFS(assembly-type flowshop)로 부르기로 한다. AFS에서 최종 조립단계를 제외하고 부품생산을 위한 전 단계만을 고려하는 단일단계모델은 주문생산방식(make-to-order)에서 고객주문 처리를 위한 일정계획문제로 해석될 수 있다. 즉, 하나의 주문 내에 다양한 제품이 포함되어져 있고, 각 제품은 독립적인 생산설비를 통해 생산 될 경우, 각 고객주문의 배달(배송)은 그 주문에 속한 모든 제품들의 생산이 완료되는 시점 이후에 가능하다. 이때 각 고객의 주문을 AFS에서의 완제품으로 해석하고, 주문에 속한 각 제품을 AFS에서의 부품으로 생각할 때 단일단계 AFS모델은 효율적인 고객주문 처리를 위한 제품일정계획문제로 해석될 수 있음을 알 수 있다. 이에 따라 최근의 관련연구들에서는 단일단계 AFS문제를 주문일정계획(order scheduling)으로 부르고 있다. 주문일정계획(단일단계 AFS)에 대해서도 최근에 많은 연구가 진행되고 있다. 먼저, Sung and Yoon(1998)은 주문들의 총 가중완료시간을 최소화하기 위한 일정계획문제가 NP-complete임을 증명하였으며, Ahmadi *et al.*(2005)은 주문에 대한 가중치를 제외하고 총 완료시간을 고려하는 보다 단순한 문제도 동일하게 NP-complete라는 것을 밝혔다. Yoon and Sung(2005)은 총 가중완료시간문제에 대해 최적 해를 도출할 수 있는 B&B 알고리즘을 제안하였으며, Wang and Cheng(2006), Leung *et al.*(2005a, 2005b), Ahmadi *et al.*(2005)은 다양한 휴리스틱들에 대한 오차한계를 증명한 바 있다 또한, Leung *et al.*(2006b)은 납기관련 함수를 목적함수로 가지는 경우의 주문일정계획문제를 다루었으며, Leung *et al.*(2006a)과 Yang and Posner(2005)는 주문에 속한 각 제품들을 독립적인 생산설비를 통해 생산하는 것이 아니고, 공통적인 병렬설비에서 생산하는 경우를 고려하였다.

본 연구의 구성은 다음과 같다 제 2장에서는 고려하고 있는 AFS에서의 일정계획문제에 대한 엄격한 정의와 분석이 제시된다. 제 3장에서는 B&B 알고리즘과 그리디 형태의 휴리스틱

이 제시된다. 제 4장에서는 개발된 휴리스틱과 B&B의 성능평가를 위한 다양한 수치실험이 이루어진다. 마지막으로 제 5장에서는 결론과 향후연구방향에 대한 논의가 이루어진다.

2. 문제 정의와 분석

본 연구에서는 AFS에서의 일정계획문제를 고려하고 있다. 대상 문제의 엄격한 정의와 분석에 사용될 기호를 기술하면 다음과 같다. 총 n 개의 작업이 수행되어야 하며, 각 작업은 m 개의 부품(또는 부분작업)과 $m+1$ 번째 조립작업으로 구성되고, 조립작업은 그 작업에 해당하는 m 개의 모든 부품가공이 완료된 이후에 시작될 수 있다. AFS는 총 $m+1$ 개의 생산설비로 구성되며 첫 번째 단계에는 m 개의 부품설비가 병렬설비(parallel machine)의 구조로 배치되어 있고 두 번째 단계에는 마지막 $m+1$ 번째 설비가 위치하여 최종 조립작업을 수행하게 된다. 각 작업의 부품 가공은 미리 할당된 부품설비에서 이루어지며 이를 기존 문헌에서는 지정설비(dedicated machine) 구조로 지칭한다(Leung *et al.*(2005a) 등을 참조). 즉, 각 작업의 k 번째 부품($k=1, 2, \dots, m$)은 부품설비 k 에서 생산되며, 작업 j 의 k 번째 부품의 가공을 위해서는 p_{jk} ($j=1, \dots, n, k=1, \dots, m$)의 가공시간이 소요된다. 또한, 작업 j 의 최종 조립작업에는 q_j ($j=1, \dots, n$)의 조립시간이 소요된다.

본 연구에서는 설명의 편의를 위해 각 작업의 k 번째 부품을 “부품 k ”라고 부르고, 부품 k 의 생산을 담당하는 설비를 “부품설비 k ”라고 부른다. 또한, 조립을 담당하는 설비를 “조립설비 $m+1$ ”으로 부르기로 한다. 또한, “작업 j ”의 “부품 k ”는 “부품설비 k ”에서만 생산되므로, 의미의 혼동이 없다면 본 연구는 “부품 k ”에 관한 기술은 생략하도록 한다. 즉, “작업 j ”의 “설비 k ”에서의 가공으로만 기술하도록 한다.

각 작업의 부품 생산은 시점 0에서부터 가공이 가능하도록 준비되어 있고, 모든 설비는 여러 개의 작업을 동시에 가공할 수 없으며, 일단 하나의 작업의 가공이 시작되면 완료될 때까지 계속된다. 각 작업의 최종 완료시간은 조립설비 $m+1$ 에서 조립이 완료되는 시점으로 모델링되며 작업 j 의 완료시간을 C_j 로 표시한다. 이때 각 작업의 완료시간에는 가중치(weight) w_j 가 할당되어 전체 n 개 작업의 총 가중완료시간은 $\sum_{j=1}^n w_j C_j$ 로 표현된다.

본 연구의 목적은 전체 작업을 완료하는데 필요한 총 가중완료시간을 최소화하는 각 설비에서의 최적 작업순서를 도출하는 것이다. 문제에 대한 이해를 돕기 위해 다음의 예제를 고려해 보자. 3개의 작업(주문)이 존재하고, 각 작업은 2개의 부품작업과 1개의 조립작업으로 구성된다. 각 작업의 가공시간 및 가중치가 <Table 1>과 같이 주어질 때 만약 작업처리순서가 작업 1, 작업 2, 작업 3의 순서로 진행된다면, <Figure 1>의 간트차트에 의해 각 작업의 완료시간은 $C_1 = 5$, $C_2 = 11$,

$C_3 = 14$ 가 되고, 총 가중완료시간은 49가 된다.

Table 1. Data for an instance

	작업 1	작업 2	작업 3
p_{j1}	2	4	2
p_{j2}	1	3	4
q_j	3	5	3
w_j	2	1	2

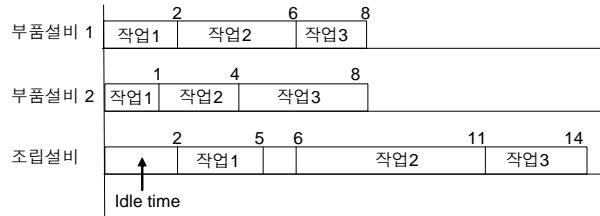


Figure 1. Gantt chart of an instance

지금까지 설명한 2단계 조립시스템에서 총 가중완료시간 최소화 문제를 편의상 AFS_w 로 부르기로 한다. 해의 도출을 위한 본격적인 분석에 들어가기 전에 AFS_w 문제가 Strongly NP-complete 문제임을 주목해야 한다. AFS에서는 임의의 부품설비 k 와 최종 조립설비 $m+1$ 간에는 2단계 흐름라인(flow-shop) 구조가 형성되며, 총 가중완료시간을 목적함수로 하는 일정계획문제는 Strongly NP-complete로 알려져 있다. 또한, 참고로 Lee *et al.*(1993)과 Potts *et al.*(1995)은 AFS에서 최종 작업 완료시간을 최소화하는 문제가 Strongly NP-complete임을 증명한 바 있다. Sung and Yoon(1998)의 연구에서는 조립단계를 고려하지 않고 전단계만을 고려하는 단일단계문제인 경우, 즉 $q_j = 0$ ($j=1, \dots, n$)의 경우에도 총 가중완료시간을 최소화하는 문제가 NP-complete임을 증명한 바 있으며, 최근에 Leung *et al.*(2005a)와 Ahmadi *et al.*(2005)은 가중치가 제외된 총 완료시간을 최소화 하는 문제도 역시 NP-complete임을 증명하였다. 따라서, 본 연구에서는 작업 수가 큰 경우에도 효율적인 해를 제공할 수 있는 휴리스틱 알고리즘의 제안과 작업 수가 크지 않은 문제에 대해서는 최적 해를 제공해 주는 B&B 알고리즘을 동시에 제안하게 된다.

다음의 성질 1과 2는 AFS_w 문제에서 최적해가 가지는 기초적인 성질들을 제시한다. 제시된 성질들을 통해 보다 효율적인 해의 탐색이 가능하게 한다.

성질 1. 최적해에서 각 부품설비는 유휴시간(idle time)없이 작업을 수행한다.

증명. 일정계획이론의 기본적인 인접작업교환(pair-wise job interchange)방법으로 쉽게 증명될 수 있으며 여기서는 구체적인 증명은 생략한다.

성질 2. 순열일정(permutation schedule)을 만족하는 적어도 하나의 최적해가 존재한다

증명. Tozkanan *et al.*(2003)의 연구에서 인접작업교환을 통해 증명된 바 있으며, 구체적인 증명은 생략한다.

성질 2는 AFS_w 문제의 최적해를 찾는 것은 $n!$ 개의 가능한 모든 순열 중에 최적의 순열을 찾는 것과 같음을 의미 한다

3. 분지한계 알고리즘과 휴리스틱 제안

3.1 분지전략(Branching strategy)

B&B트리의 각 노드는 전체 작업 중 일부로만 구성되는 부분순열(partial sequence)에 일대일로 대응된다. 이때 부분순열은 전체 순열에서 앞부분에 위치하게 된다. 분지단계에서는 분지할 하나의 노드(부분순열)를 선택하고 이 부분순열에 포함되지 않은 작업들 중 한 개의 작업을 이 부분순열의 바로 뒤에 삽입함으로써 새로운 자식노드가 발생하게 된다. 따라서 만약 어떤 노드가 a 개의 작업들로 구성되는 부분순열에 해당한다면 이 노드를 분지함으로써 총 $(n-a)$ 개의 새로운 자식노드가 발생하게 된다. 새롭게 만들어지는 자식노드는 $(a+1)$ 개의 작업들로 구성되어진다.

분지할 노드의 선택은 깊이우선탐색(depth-first search)을 사용한다. 이는 탐색트리에서 가장 큰 부분순열을 가진 노드가 분지노드로 선택됨을 의미한다. 만일 동물이 발생한 경우에는 그중에서 하한 값(lower bound)이 가장 작은 노드가 선택된다.

3.2 한계규칙(Bounding rule)

한계규칙은 각 노드의 상한과 하한을 이용하여 현재 탐색중인 노드의 제거(fathoming) 여부를 결정하고, 해의 수렴을 촉진시킨다. 초기 상한(initial upper bound)으로는 3.4절에서 제시하는 휴리스틱 해를 사용한다. 이후 탐색에서 더 좋은 가능해가 얻어지면 상한 값은 갱신된다.

또한, 하한으로는 세 개의 하한 값이 사용된다. 이의 도출과 설명을 위해 다음의 기호들이 사용된다.

π : 전체 작업들의 부분집합으로 구성되는 부분순열

n_π : π 에 포함된 작업의 수,

T_k : 설비 k 에서 π 에 속한 마지막 작업의 완료시점 단, $k = 1, \dots, m+1$,

$\tilde{\pi}$: 집합 π 에 속하지 않은 나머지 작업들로 구성된 임의의 부분순열,

S : 부분순열 π 가 앞부분에 위치하고, 그 뒤를 이어 $\tilde{\pi}$ 가 위치하는 임의의 전체순열,

x_{jt} : 작업 j 가 t 번째 생산순서에 위치하면 1의 값을 가지고, 그렇지 않다면 0의 값을 가지는 0-1변수, 단, $j \in \tilde{\pi}$,

$$t = n_\pi + 1, \dots, n,$$

$$Q_{jk}: \text{부품설비 } k \text{에서 작업 } j \text{의 완료시점 단, } j \in \tilde{\pi}, k = 1, \dots, m.$$

작업들의 순서가 정해진 부분순열 π 로 시작하는 임의의 전체순열 S 의 총 가중완료시간 $TC(S)$ 는 다음과 같이 표현된다.

$$TC(S) = TC(\pi) + TC(\tilde{\pi})$$

여기서, $TC(\pi)$ 와 $TC(\tilde{\pi})$ 는 각각 π 와 $\tilde{\pi}$ 의 가중완료시간의 합이다. $TC(\pi)$ 는 부분순열 π 에 대해 고정된 값으로 쉽게 계산할 수 있다. 한편, $TC(\tilde{\pi})$ 는 다음과 같은 수식으로 표현된다.

$$TC(\tilde{\pi}) = \sum_{j \in \tilde{\pi}} w_j C_j \quad (1)$$

본 연구에서는 $TC(\tilde{\pi})$ 에 대한 하한을 구하기 위해 다음과 같은 수리모형(mathematical formulation)을 제안한다.

$$\text{Min } \sum_{j \in \tilde{\pi}} w_j C_j$$

subject to

$$C_j \geq Q_{jk} + q_j, \text{ for } j \in \tilde{\pi}, k = 1, \dots, m \quad (2)$$

$$Q_{jk} \geq \sum_{i \in \tilde{\pi}} Q_{ik} x_{it-1} + p_{jk} - M(1 - x_{jt}), \text{ for } j \in \tilde{\pi}, k = 1, \dots, m, t = n_\pi + 1, \dots, n \quad (3)$$

$$C_j \geq \sum_{i \in \tilde{\pi}} C_i x_{it-1} + q_j - M(1 - x_{jt}), \text{ for } j \in \tilde{\pi}, t = n_\pi + 1, \dots, n \quad (4)$$

$$Q_{jk} \geq T_k + p_{jk}, \text{ for } j \in \tilde{\pi}, k = 1, \dots, m \quad (5)$$

$$C_j \geq T_{m+1} + q_j, \text{ for } j \in \tilde{\pi} \quad (6)$$

$$\sum_{t=n_\pi+1}^n x_{jt} = 1, \text{ for } j \in \tilde{\pi} \quad (7)$$

$$\sum_{j \in \tilde{\pi}} x_{jt} = 1, \text{ for } t = n_\pi + 1, \dots, n \quad (8)$$

$$x_{jt} \in \{0, 1\}, C_j, Q_{jk} \in Z^+ \cup \{0\},$$

$$\text{for } j \in \tilde{\pi}, k = 1, \dots, m, t = n_\pi + 1, \dots, n$$

제약식 (2)는 조립설비 $m+1$ 에서 작업 j 의 조립은 부품설비들 $1, \dots, m$ 에서의 가공이 완료된 후에 시작될 수 있음을 의미한다. 제약식 (3)과 (4)는 설비 k ($k = 1, \dots, m+1$)에서의 작

업 j 의 가공은 해당 설비 k 에서 앞선 작업들의 가공이 완료된 후에 시작될 수 있음을 의미한다 여기서 M 은 임의의 큰 값을 나타낸다. 제약식 (5)와 (6)은 설비 k ($k=1, \dots, m+1$)에서 작업 j 의 가공은 해당 설비 k 에서 부분순열 π 에 속한 모든 작업들의 가공이 완료된 후에 시작될 수 있음을 의미한다 제약식 (7)과 (8)은 모든 작업들은 반드시 할당되어야 함을 의미한다.

본 연구는 위의 수리모형에서 제약식의 일부를 완화하여 세 가지 하한값(LB_1, LB_2, LB_3)을 제안한다. 첫 번째 하한 LB_1 은 다음과 같이 유도된다. 위의 수리모형에서 제약식 (3)을 완화하면 변수 Q_{jk} ($j \in \tilde{\pi}, k=1, \dots, m$)는 $T_k + p_{jk}$ 의 값으로 고정된다. 이를 활용하면 식 (1)의 $TC(\tilde{\pi})$ 는 다음과 같이 표현될 수 있다.

$$TC(\tilde{\pi}) \geq \sum_{j=n_{\pi}+1}^n w_{[j]} \left(\max \{ T_{m+1}, \max_{1 \leq k \leq m} (\min_{j \in \tilde{\pi}} Q_{jk}) \} + \sum_{i=n_{\pi}+1}^j q_{[i]} \right) \quad (9)$$

여기서, $[i]$ 는 전체순열 S 에서 i 번째 순서에 위치하는 작업을 의미한다. 위의 식을 다시 쓰면 다음과 같다.

$$TC(\tilde{\pi}) \geq \max \{ T_{m+1}, \max_{1 \leq k \leq m} (\min_{j \in \tilde{\pi}} Q_{jk}) \} \sum_{j=n_{\pi}+1}^n w_{[j]} + \sum_{j=n_{\pi}+1}^n w_{[j]} \left(\sum_{i=n_{\pi}+1}^j q_{[i]} \right) \quad (10)$$

여기서 T_{m+1} 과 Q_{jk} 가 고정되므로 위의 식의 첫 번째 항은 상수가 되고, 두 번째 항은 $\tilde{\pi}$ 에 속하는 작업들을 조립작업시간 q_j 에 대해 WSPT(weighted shortest processing time; q_j/w_j)순으로 정렬했을 때 최소화 될 수 있다. 따라서 $WSPT_{m+1}(\tilde{\pi})$ 가 $n-n_{\pi}$ 개의 작업집합 $\{q_i, w_i | i \in \tilde{\pi}\}$ 에 대해 WSPT순서의 결과로 발생하는 총 가공완료시간을 나타낸다고 할 때 첫 번째 하한 LB_1 은 다음과 같이 정의된다.

$$LB_1 = TC(\pi) + \max \{ T_{m+1}, \max_{1 \leq k \leq m} (\min_{j \in \tilde{\pi}} Q_{jk}) \} \sum_{j=n_{\pi}+1}^n w_{[j]} + WSPT_{m+1}(\tilde{\pi}) \quad (11)$$

다음으로 LB_2 과 LB_3 의 유도과정을 설명한다. 수리모형에서 제약식 (4)와 (6)을 완화하면, 전체순열 S 에서, 변수 $Q_{[j]k}$ ($j=n_{\pi}+1, \dots, n, k=1, \dots, m$)는 다음과 같이 표현된다.

$$Q_{[j]k} = T_k + \sum_{i=n_{\pi}+1}^j p_{[i]k}, \quad \text{for } j=n_{\pi}+1, \dots, n, k=1, \dots, m \quad (12)$$

또한, 변수들 $C_{[j]}$ ($j=n_{\pi}+1, \dots, n$)는 다음과 같이 표현될 수 있다.

$$C_{[j]} = \max_{1 \leq k \leq m} \{ Q_{[j]k} \} + q_{[j]}, \quad \text{for } j=n_{\pi}+1, \dots, n \quad (13)$$

따라서, $TC(\tilde{\pi})$ 는 다음과 같이 표현된다.

$$TC(\tilde{\pi}) = \sum_{j=n_{\pi}+1}^n w_{[j]} (\max_{1 \leq k \leq m} \{ Q_{[j]k} \} + q_{[j]}) = \sum_{j=n_{\pi}+1}^n w_{[j]} \max_{1 \leq k \leq m} \{ Q_{[j]k} \} + \sum_{j=n_{\pi}+1}^n w_{[j]} q_{[j]} \quad (14)$$

식 (12)와 식 (14)를 결합하면, $TC(\tilde{\pi})$ 는 다음과 같이 표현된다.

$$TC(\tilde{\pi}) = \sum_{j=n_{\pi}+1}^n w_{[j]} \max_{1 \leq k \leq m} \left\{ T_k + \sum_{i=n_{\pi}+1}^j p_{[i]k} \right\} + \sum_{j=n_{\pi}+1}^n w_{[j]} q_{[j]} \geq \max_{1 \leq k \leq m} \left\{ \sum_{j=n_{\pi}+1}^n w_{[j]} \left(T_k + \sum_{i=n_{\pi}+1}^j p_{[i]k} \right) \right\} + \sum_{j=n_{\pi}+1}^n w_{[j]} q_{[j]} = \max_{1 \leq k \leq m} \left\{ T_k \sum_{j=n_{\pi}+1}^n w_{[j]} + \sum_{j=n_{\pi}+1}^n w_{[j]} \sum_{i=n_{\pi}+1}^j p_{[i]k} \right\} + \sum_{j=n_{\pi}+1}^n w_{[j]} q_{[j]} \quad (15)$$

식 (15)의 마지막 항과 중괄호안의 첫 번째 항은 상수이다. 중괄호안의 두 번째 항은 $\tilde{\pi}$ 에 속하는 작업들을 설비 k ($k=1, \dots, m$)에서 작업시간 p_{jk} 에 대해 WSPT($=p_{jk}/w_j$)순으로 정렬할 때 최소화 될 수 있다. 따라서 $WSPT_k(\tilde{\pi})$ ($k=1, \dots, m$)가 $n-n_{\pi}$ 개의 작업집합 $\{p_{ik}, w_i | i \in \tilde{\pi}\}$ 에 대해 WSPT순서의 결과로 발생하는 총 가공완료시간을 나타낸다고 할 때 $\sum_{j=n_{\pi}+1}^n w_{[j]} \sum_{i=n_{\pi}+1}^j p_{[i]k} \geq WSPT_k(\tilde{\pi})$ 의 관계가 성립하므로 식 (15)로부터 두 번째 하한 LB_2 는 다음과 같이 도출될 수 있다.

$$LB_2 = TC(\pi) + \max_{1 \leq k \leq m} \left\{ T_k \sum_{j \in \tilde{\pi}} w_j + WSPT_k(\tilde{\pi}) \right\} + \sum_{j \in \tilde{\pi}} w_j q_j \quad (16)$$

세 번째 하한을 구하기 위해서 식 (15)를 다음과 같이 다시 표현한다.

$$\text{식 (15)} \geq \frac{1}{m} \sum_{k=1}^m \left\{ T_k \sum_{j=n_{\pi}+1}^n w_{[j]} + \sum_{j=n_{\pi}+1}^n w_{[j]} \sum_{i=n_{\pi}+1}^j p_{[i]k} \right\} + \sum_{j=n_{\pi}+1}^n w_{[j]} q_{[j]} = \frac{1}{m} \sum_{k=1}^m T_k \sum_{j=n_{\pi}+1}^n w_{[j]} + \frac{1}{m} \sum_{j=n_{\pi}+1}^n w_{[j]} \sum_{i=n_{\pi}+1}^j p_{[i]k} + \sum_{j=n_{\pi}+1}^n w_{[j]} q_{[j]} \quad (17)$$

위 식의 첫 번째 항과 세 번째 항은 상수이고, 두 번째 항은 $\tilde{\pi}$ 에 속하는 작업들을 각각의 작업시간의 합 $\sum_{k=1}^m p_{jk}$ 에 대해 $WSPT(=\sum_{k=1}^m p_{jk} / w_j)$ 순으로 정렬했을 때 최소화 될 수 있다. 즉, $WSPT_{\Sigma}(\tilde{\pi})$ 가 $n - n_{\tilde{\pi}}$ 개의 작업집합 $\left\{ \sum_{k=1}^m p_{ik}, w_i | i \in \tilde{\pi} \right\}$ 에 대한 $WSPT$ 순서의 결과로 발생하는 총 가중완료시간을 나타낸다고 할 때 마지막 세 번째 하한 LB_3 은 다음과 같이 정의된다.

$$LB_3 = TC(\pi) + \frac{1}{m} \sum_{k=1}^m T_k \sum_{j \in \tilde{\pi}} w_j + \frac{1}{m} WSPT_{\Sigma}(\tilde{\pi}) + \sum_{j \in \tilde{\pi}} w_j f_j \quad (18)$$

본 연구의 B&B 알고리즘에서는 도출된 세 가지 하한 LB_1, LB_2, LB_3 중에서 가장 우수한 것을 각 노드별 최종 하한으로 사용한다. 즉 각 노드에서의 최종 하한값 LB 는 다음과 같이 계산된다.

$$LB = \max\{LB_1, LB_2, LB_3\}$$

3.3 노드제거규칙(Fathoming rule)

노드제거규칙은 Tozkanpan *et al.*(2003)에서 제시된 다음의 성질 3이 사용된다. 먼저, 두 가지 부분순열 $\pi = (\pi(1), \dots, \pi(s))$ 와 $\bar{\pi} = (\pi(1), \dots, \pi(s-2), \pi(s), \pi(s-1))$ 를 고려하자. 즉, $\bar{\pi}$ 는 부분순열 π 에서 마지막 두 작업의 순서를 바꾼 것이다.

성질 3. (Tozkanpan *et al.*, 2003) 만일 다음의 관계들이 만족된다면, $\bar{\pi}$ 가 π 보다 더 우월하다.

- 1) $TC(\pi) \geq TC(\bar{\pi})$
- 2) $TC(\pi) - TC(\bar{\pi}) \geq \sum_{r \in \pi} w_r (L(\bar{\pi}) - L(\pi))$

여기서, $L(\pi)$ 과 $L(\bar{\pi})$ 는 π 와 $\bar{\pi}$ 각각에 대한 조립설비 $m+1$ 에서의 마지막 작업의 조립완료시점을 의미한다

따라서 만일 성질 3이 만족할 경우 부분순열 π 에 대응하는 분지노드는 제거된다.

3.4 상한한계절차(Upper bound procedure)

본 연구에서 고려하는 일정계획문제는 Strongly NP-complete이므로 크기가 큰 문제에 대해 빠른 계산으로 효과적인 해를 제시할 수 있는 휴리스틱의 개발이 중요한 과제라 할 수 있다. 또한 본 절에서 제공하는 휴리스틱은 그 자체의 중요성 뿐만 아니라 앞에서 제시된 B&B 알고리즘에서 루트노드의 상

한 값을 제공함으로써 노드 탐색을 크게 줄이는 효과를 제공한다.

본 연구에서는 총 세 가지 휴리스틱 $G1, G2, G3$ 을 제안한다. 첫 번째 휴리스틱을 설명하면 다음과 같다. $T_k (k=1, \dots, m+1)$ 를 부품설비 k 에서 선행하는 작업들의 완료시간이라 할 때, 아직 가공순서가 정해지지 않은 작업 j 를 그 뒤에 할당하게 되면 작업 j 의 작업(조립)완료시간 C_j 는 $\max\{\max_{1 \leq k \leq m} \{T_k + p_{jk}\}, T_{m+1}\} + q_j$ 로 결정된다. $WSPT$ 의 원리를 활용하여, 다음의 값이 가장 작은 작업을 선택하여 다음 위치에 할당한다.

$$\frac{1}{w_j} (\text{작업 } j \text{의 조립완료시간} - \text{이전작업의 조립완료시간}) = \frac{1}{w_j} (C_j - T_{m+1}).$$

이를 활용한 휴리스틱 $G1$ 은 다음과 같이 표현된다.

그리디 휴리스틱 $G1$

단계 0: $\tilde{\pi} = \{1, 2, \dots, n\}$, $T_k = 0 (k=1, \dots, m+1)$.

단계 1: $\tilde{\pi}$ 에 속하는 작업들 중에서

$$\min_{j \in \tilde{\pi}} \left[\frac{1}{w_j} (\max\{\max_{1 \leq k \leq m} \{T_k + p_{jk}\}, T_{m+1}\} + q_j - T_{m+1}) \right]$$

값을 가지는 작업 x 를 선택한다.

단계 2: 작업 x 를 집합 $\tilde{\pi}$ 에서 제거하고, 시점 $T_k (k=1, \dots, m)$ 를 $(T_k + p_{xk})$ 의 값으로, T_{m+1} 을 $(\max\{\max_{1 \leq k \leq m} \{T_k\}, T_{m+1}\} + q_j)$ 의 값으로 업데이트 한다.

단계 3: $\tilde{\pi}$ 가 공집합이면 종료하고, 그렇지 않으면 단계 1로 간다.

다음으로 휴리스틱 $G2$ 와 $G3$ 는 $G1$ 의 단계 1에서 작업선택의 기준값을 변경함으로써 얻어진다 만일 부품생산단계가 조립단계에 비해 상대적으로 병목공정(bottleneck process)일 경우, 부품생산단계의 완료시점을 중심으로 한 작업선택이 유효하다. 이런 관점에서의 새로운 작업선택의 기준값은 다음과 같다.

$$\frac{1}{w_j} (\text{작업 } j \text{의 부품생산완료시점} - \text{이전작업의 부품생산완료시점})$$

이를 활용한 두 번째 휴리스틱 $G2$ 는 다음과 같다.

그리디 휴리스틱 $G2$

단계 0: $\tilde{\pi} = \{1, 2, \dots, n\}$, $T_k = 0 (k=1, \dots, m+1)$.

단계 1: $\tilde{\pi}$ 에 속하는 작업들 중에서

$$\min_{j \in \tilde{\pi}} \left[\frac{1}{w_j} (\max_{1 \leq k \leq m} \{T_k + p_{jk}\} - \max_{1 \leq k \leq m} \{T_k\}) \right]$$

값을 가지는 작업 x 를 선택한다.

단계 2, 3: $G1$ 과 동일하다.

휴리스틱 $G1$ 은 작업의 조립단계에 보다 가중치를 부여하는 휴리스틱이고, $G2$ 는 부품생산단계에 보다 가중치를 부여하는 휴리스틱이다. 마지막으로 부품생산단계와 조립단계에 비슷한 가중치를 부여하는 휴리스틱이 $G3$ 이다. $G3$ 에서 사용하는 작업선택의 기준값은 다음과 같다.

$$\frac{1}{w_j}(\text{작업 } j \text{의 부품생산완료시점} - \text{이전작업의 부품생산완료시점})$$

이를 활용한 마지막 휴리스틱 $G3$ 는 다음과 같다.

그리디 휴리스틱 $G3$

단계 0: $\tilde{\pi} = 1, 2, \dots, n, T_k = 0 (k = 1, \dots, m + 1)$.

단계 1: $\tilde{\pi}$ 에 속하는 작업들 중에서

$$\min_{j \in \tilde{\pi}} \left[\frac{1}{w_j} (\max\{\max_{1 \leq k \leq m} \{T_k + p_{jk}\} - T_{m+1}, 0\}) \right]$$

값을 가지는 작업 x 를 선택한다.

단계 2, 3: $G1$ 과 동일하다.

본 연구에서 제안된 세 가지 휴리스틱의 목적값을 UB_1, UB_2, UB_3 라고 할 때, 이 중에서 가장 우수한 것을 B&B의 초기 상한값으로 사용한다. 즉, 분지한계 알고리즘의 초기상한 UB 는 다음과 같다.

$$UB = \min\{UB_1, UB_2, UB_3\}$$

3.5 B&B 알고리즘의 전체절차

본 절에서는 앞에서 제시한 분지규칙, 한계규칙, 상한한계절차들을 활용하여 전체적인 B&B 알고리즘을 표현한다. 설명에 사용될 기호들은 다음과 같이 정의된다

- r : 전체 분지트리에서 해당 노드의 수준(level),
- π_r : 수준 r 에서 이미 순서가 정해진 작업들의 집합,
- $\tilde{\pi}_r$: 수준 r 에서 아직 순서가 정해지지 않은 작업들의 집합,
- $LB_r(j)$: 수준 r 의 노드에서 작업 j 로 분지한다고 가정할 때의 하한 값,
- Z^* : 상한 값.

B&B 알고리즘

단계 0: (초기화).

$r = 0, \pi_r = \emptyset, \tilde{\pi}_r = \{1, 2, \dots, n\}$ 로 각각 초기화한다.

3.4절의 휴리스틱을 적용하여 얻어진 초기 상한 값 UB 를 Z^* 에 저장한다.

단계 1: (하한 계산).

$j \in \tilde{\pi}_r$ 인 모든 작업에 대해 $LB_r(j)$ 값을 계산한다.

단계 2: (분지단계)

2.1 만일 $|\tilde{\pi}_r| = 1$ 이면, 단계 4로 이동한다.

2.2 집합 $\tilde{\pi}_r$ 에 속한 모든 작업 j 에 대해 $LB_r(j) = \infty$ 이면, 단계 5로 이동한다.

2.3 집합 $\tilde{\pi}_r$ 에 속한 작업들 중에서 가장 작은 $LB_r(j)$ 값을 가지는 작업 k 를 선택하고, 다음과 같이 각 항목들을 갱신한다.

$$r \leftarrow (r + 1), \pi_r = \pi_{r-1} \cup \{k\}, \tilde{\pi}_r = \tilde{\pi}_{r-1} - \{k\}.$$

단계 3: (우월성질 및 하한 검사).

3.1 만일 $LB_r(j) \geq Z^*$ 을 만족하면, 단계 5로 이동한다.

3.2 작업 x, y 가 집합 π_r 에서 가장 마지막에 위치하고 $x \rightarrow y$ 순서로 처리된다고 가정하자. 이때 성질 3에 의해 $y \rightarrow x$ 순서가 우월할 경우에는, 단계 5로 이동한다.

3.3 단계 1로 이동한다.

단계 4: (상한값 갱신).

열거트리의 마지막 레벨에 도달한 경우에는 발견된 가능해의 목적함수 값 \bar{Z} 를 계산한다. 만약 $Z^* > \bar{Z}$ 라면, Z^* 의 값을 \bar{Z} 로 갱신한다.

단계 5: (분지완료단계).

현재의 노드는 더 이상 탐색할 필요가 없다.

$r \leftarrow (r - 1), LB_r(j) = \infty$ 로 갱신한다.

만일 $r \geq 0$ 이면, 단계 2로 이동하고, 그렇지 않으면, 알고리즘을 종료한다(이때, 최적해의 값은 Z^* 단계이다).

4. 성능평가

본 장에서는 수치실험을 통해 제안된 B&B 알고리즘의 성능을 Tozkanpan *et al.*(2003)의 결과와 비교하게 된다. 알고리즘들은 C언어로 프로그래밍 되었고, Pentium IV 2.4 GHz PC에서 실행되었다. 먼저 평가에 사용된 실험문제들의 생성방법은 다음과 같다. 납기 가중치(w_i)는 $U(1,10)$ 으로 부터 발생시킨다. 여기서, $U(a,b)$ 는 a 와 b 를 모수로 가지는 이산형 균등분포를 의미한다. 각 작업의 가공시간은 다음<Table 2>와 같이 데이터 종류에 따라 서로 다른 분포로 부터 생성된다.

Table 2. Processing Time Distributions

데이터 유형	부품단계의 가공시간 분포	조립단계의 가공시간 분포
A	U(1,100)	U(1,100)
B	U(1,80)	U(20,100)
C	U(20,100)	U(1,80)

표에서 3종류의 데이터는 부품단계와 조립단계에 대한 각

각의 병목상황을 반영한 것이다 즉, 데이터 유형 A는 부품단계와 조립단계에 비슷한 작업부하가 할당된 경우이고, 유형 B는 조립단계를 병목공정으로 지정한 경우이고 유형 C는 반대로 부품단계가 병목공정일 가능성이 크다고 할 수 있다

먼저 데이터 유형 A에 대해 실험을 실시하였다. 작업 수(n)를 9종류(10, 12, 14, 16, 18, 20, 22, 24, 26), 부품설비의 수(m)를 3종류(2, 5, 10)에서 선택하여 총 27가지의 문제조합에 대해 각각 30문제씩, 총 810문제를 임의로 생성하였다. 데이터 유형 A에 대한 성능비교 실험의 결과는 <Table 3>에 요약되어 있다. 표에서 “BB1”은 Tozkanpan *et al.*의 B&B 알고리즘의 결과를 의미하고, “BB2”는 본 연구에서 제안된 B&B의 결과를 나타낸다. “npl”은 해당 알고리즘이 각 셀의 30개 문제 중에서 최적 해를 600초(CPU시간)안에 찾아낸 문제의 수를 표시한 것이다. “노드 수”는 각 B&B 트리에서 생성된 총 노드 수를 의미한다.

데이터 유형 B에 대한 성능비교 실험의 결과는 <Table 4>에 요약되어 있다. 작업 수를 8종류, 부품설비의 수를 3종류에서 선택하여 총 24개의 가능한 조합에 대해 각각 30문제씩, 총

720문제를 임의로 생성하였다.

데이터 유형 C에 대한 결과는 <Table 5>에 요약되어 있다. 작업 수와 부품설비 수의 24가지 조합에서 총 720문제를 생성하여 실험을 진행하였다. 이상의 실험결과에서 볼 수 있듯이 모든 데이터유형과 문제조합에 대해 BB2가 BB1보다 항상 우수한 결과를 보임을 알 수 있다. 특히, 부품단계가 병목인 경우(데이터 유형 C)와 서로 대등한 경우(데이터 유형 A)에서 그 격차가 더 커짐을 알 수 있다.

Tozkanpan *et al.*(2003)은 개별 부품생산설비들에 대해 독립적인 WSPT순서를 구하거나, 전체 부품생산설비를 가상적인 단일기계로 집단화(aggregation)하고 이에 대해 WSPT 순서를 구하는 방식으로 2가지 휴리스틱을 제안한 바 있다. 본 연구는 3.4절에서 제시한 바와 같이 각 단계에서 가장 유리하다고 판단이 되는 작업을 한 개씩 그리디 방식으로 선택하는 3가지 휴리스틱을 제안하였다. <Table 6>은 본 연구에서 제안된 휴리스틱과 Tozkanpan *et al.*에서 제안된 휴리스틱과의 성능 비교실험을 수행한 결과를 보여주고 있다. 표에서 “UB1”은 Tozkanpan *et al.*의 휴리스틱의 결과를 의미하고 “UB2”는 본 연구의 결과

Table 3. Experimental results for data-type A

부품설비 수	작업수	BB1					BB2				
		노드수		계산시간(초)		npl	노드수		계산시간(초)		npl
		평균	최대	평균	최대		평균	최대	평균	최대	
2	10	188.80	1238	0.01	0.05	30	70.67	380	0.00	0.01	30
	12	748.57	2877	0.05	0.19	30	272.57	2383	0.01	0.13	30
	14	4281.87	41557	0.35	3.28	30	1009.67	4968	0.07	0.33	30
	16	34697.47	282204	3.42	26.48	30	3655.77	52524	0.37	4.95	30
	18	205855.33	1579065	25.11	184.31	30	9311.53	35929	1.32	4.14	30
	20	(*)	(*)	(*)	(*)	26	94227.53	640599	17.23	117.27	30
	22	(*)	(*)	(*)	(*)	21	260097.57	2272673	59.08	505.11	30
	24	(*)	(*)	(*)	(*)	10	988178.83	9343527	259.54	2481.11	28
5	10	334.13	2660	0.02	0.16	30	239.10	2088	0.01	0.11	30
	12	2139.33	15971	0.17	1.06	30	1043.50	8951	0.09	0.64	30
	14	10809.63	130520	1.19	12.73	30	2949.60	15352	0.37	1.94	30
	16	47607.47	201808	7.08	30.16	30	16517.07	82672	2.76	13.39	30
	18	541865.03	5771015	96.45	1007.41	28	68020.13	485164	15.09	108.00	30
	20	(*)	(*)	(*)	(*)	23	524971.87	7006070	158.82	1960.94	28
	22	(*)	(*)	(*)	(*)	11	1200893.53	9543717	500.13	3123.41	23
	24	(*)	(*)	(*)	(*)	7	(*)	(*)	(*)	(*)	11
10	10	558.60	3309	0.05	0.25	30	275.13	1503	0.03	0.30	30
	12	2186.47	22312	0.26	1.98	30	782.30	2571	0.12	0.38	30
	14	9584.97	65953	1.70	10.30	30	4881.73	46002	1.00	8.47	30
	16	125493.40	1542494	28.33	326.81	30	33374.03	314516	9.10	78.17	30
	18	442301.80	4287165	133.02	1129.41	28	90417.50	883838	35.82	340.97	30
	20	(*)	(*)	(*)	(*)	17	(*)	(*)	(*)	(*)	22

* 3600초(cpu) 안에 풀 수 없는 문제가 존재하여 통계 값을 얻을 수 없는 경우임

Table 4. Experimental results for data-type B

부품설 비수	작업수	BB1					BB2				
		노드수		계산시간(초)		npl	노드수		계산시간(초)		npl
		평균	최대	평균	최대		평균	최대	평균	최대	
2	15	441.37	2519	0.06	0.38	30	160.03	1776	0.02	0.22	30
	18	711.13	1816	0.14	0.53	30	227.53	1405	0.06	0.38	30
	21	1713.57	6949	0.41	2.78	30	325.33	4546	0.12	1.89	30
	24	4496.50	27321	1.31	6.06	30	422.77	3585	0.23	2.02	30
	27	10215.27	123817	4.27	41.81	30	1062.90	21405	0.89	18.83	30
	30	46875.33	952703	41.19	893.25	29	8836.07	201655	9.82	222.80	30
	33	20049.10	151611	16.23	140.76	30	914.80	3737	1.43	6.38	30
	36	(*)	(*)	(*)	(*)	28	(*)	(*)	(*)	(*)	29
5	15	491.23	6735	0.06	0.91	30	211.50	4621	0.03	0.67	30
	18	1556.13	18567	0.36	5.06	30	783.80	12508	0.22	3.39	30
	21	12991.00	302575	4.53	112.89	30	11356.97	312060	4.52	123.67	30
	24	233093.73	6782204	77.86	2245.00	29	132034.7	3884622	61.26	1781.23	29
	27	37395.13	562446	29.20	492.64	30	22753.33	425196	21.66	413.88	30
	30	50904.13	478768	40.05	392.31	30	19738.33	312527	29.26	428.78	30
	33	(*)	(*)	(*)	(*)	28	(*)	(*)	(*)	(*)	28
10	15	384.30	1013	0.08	0.31	30	155.70	786	0.05	0.38	30
	18	884.70	2284	0.22	0.50	30	190.73	797	0.09	0.42	30
	21	2301.60	13573	1.02	9.13	30	754.97	8782	0.58	6.67	30
	24	9798.00	174469	8.47	180.48	30	6245.33	158720	7.48	189.08	30
	27	7272.17	40050	6.15	39.53	30	1968.17	18805	3.38	33.59	30
	30	65331.00	1450944	133.77	3189.13	29	54337.97	1413323	131.72	3444.53	29
	33	(*)	(*)	(*)	(*)	29	38580.80	894015	119.32	2767.11	29
	36	(*)	(*)	(*)	(*)	22	(*)	(*)	(*)	(*)	24

Table 5. Experimental results for data-type C

부품설 비수	작업수	BB1					BB2				
		노드수		계산시간(초)		npl	노드수		계산시간(초)		npl
		평균	최대	평균	최대		평균	최대	평균	최대	
2	12	1040.37	6114	0.07	0.36	30	127.27	549	0.01	0.03	30
	14	12912.57	72171	0.87	4.45	30	796.60	7027	0.06	0.45	30
	16	44069.13	334742	4.37	37.22	30	1792.43	18907	0.20	2.17	30
	18	658008.13	12553555	61.89	1118.36	29	7836.27	90987	1.02	13.38	30
	20	1594234.63	18132575	187.29	1711.61	28	15095.67	112690	2.33	14.89	30
	22	(*)	(*)	(*)	(*)	23	23438.10	259469	4.36	46.03	30
	24	(*)	(*)	(*)	(*)	11	175954.87	2267833	50.07	560.91	30
	26	(*)	(*)	(*)	(*)	1	(*)	(*)	(*)	(*)	24
5	12	761.37	4700	0.11	0.48	30	230.80	1240	0.02	0.14	30
	14	4294.60	26281	0.90	5.63	30	1309.40	11865	0.21	1.73	30
	16	15840.43	79455	4.40	20.70	30	2472.30	13194	0.54	2.67	30
	18	195911.87	1435062	62.22	431.78	30	20613.30	151328	5.13	35.44	30
	20	(*)	(*)	(*)	(*)	23	129731.53	1369256	41.26	436.47	30
	22	(*)	(*)	(*)	(*)	16	788113.60	10064792	249.34	1623.03	25
	24	(*)	(*)	(*)	(*)	3	(*)	(*)	(*)	(*)	21
10	12	1434.43	16350	0.27	2.80	30	445.90	6019	0.07	0.88	30
	14	7660.37	44852	2.38	12.48	30	1619.80	6940	0.43	1.74	30
	16	63519.50	1247520	20.71	342.89	30	7020.83	49770	2.45	16.78	30
	18	279930.37	2152355	122.01	736.81	28	28810.43	215206	12.59	85.14	30
	20	(*)	(*)	(*)	(*)	22	336704.30	5345453	169.05	2526.14	27
	22	(*)	(*)	(*)	(*)	13	768519.10	4965060	595.00	3369.67	23
	24	(*)	(*)	(*)	(*)	3	(*)	(*)	(*)	(*)	14

를 나타낸다. 표에서 “Gap1”은 $\frac{(UB - Opt)}{Opt} \times 100$ 의 값을 가진다. 여기서, “UB”는 각 휴리스틱으로부터 얻어진 목적함수 값을 의미하고, “Opt”는 B&B를 통해 얻은 최적해의 목적함수 값을 나타낸다. 표의 결과에서 알 수 있듯이 본 연구에서 제안된 휴리스틱의 경우가 전반적으로 보다 더 효과적인 해를 제공하고 있음을 알 수 있다.

Table 6. Experimental results for heuristics

작업수	UB1의 Gap1(%)		UB2의 Gap1(%)	
	평균	최대	평균	최대
유형 A				
10	4.46	14.42	2.56	10.99
12	5.42	16.80	3.56	16.59
14	6.45	16.27	4.60	13.22
16	7.40	19.57	4.50	8.79
유형 B				
15	5.65	13.04	4.59	13.04
18	4.85	11.75	3.74	8.64
21	4.96	12.13	3.98	11.40
24	5.56	11.27	3.80	11.09
27	5.89	13.59	4.40	13.59
30	6.58	12.91	5.21	10.35
유형 C				
12	1.60	5.86	0.43	2.08
14	2.15	5.91	0.65	2.36
16	2.49	6.39	0.56	2.14
18	2.76	5.48	0.82	3.57

5. 결론 및 향후연구과제

본 연구는 직렬과 병렬이 혼재된 2단계 조립시스템에서의 일정계획 문제를 다루었다. 기존연구의 대부분이 최종완료시간을 최소화하는 문제에 국한되어 있는데 반해 본 연구는 보다 복잡한 총 가중완료시간을 목적함수로 채택하였다. 효율적인 B&B 알고리즘을 위한 새로운 하한값들이 제안되었으며, 효과적인 해를 제공하는 몇 가지 휴리스틱이 개발되었다. 또한 제안된 알고리즘들이 기존 연구결과에 비해 성능이 우수하다는 것을 다양한 수치실험을 통해 검증하였다.

목적함수를 작업완료시간의 합이나 납기관련 함수들로 다양화 할 경우 2단계 조립시스템에서는 많은 연구 과제를 도출할 수 있다. 하지만, 2단계 조립시스템은 부분적으로 2단계 흐름라인들을 결합한 것이므로, 2단계 조립시스템에 대한 후속 연구는 흐름라인에 대한 기존 연구결과에 대한 고찰로부터 출발해야 할 것으로 사료된다.

참고문헌

Ahmadi, R., Bagchi, U., and Roemer, T. A. (2005), Coordinated Scheduling of Customer Orders for Quick Response, *Naval Research Logistics*, **52**, 493-512.

Baker, K. R. (1974), *Introduction to Sequencing and Scheduling*, John Wiley & Sons.

Baker, K. R. and Powell, S. G. (1993), A Predictive Model for the Throughput of Simple Assembly Systems, *European Journal of Operational Research*, **81**, 336-345.

Baker, K. R., Powell, S. G., and Pyke, D. F. (1993), Optimal Allocation of Work in Assembly Systems, *Management Science*, **39**, 101-106.

Bhat, U. N. (1986), Finite Capacity Assembly-like Queues, *Queueing Systems*, **1**, 85-101.

Duda, A. and Czachorski, T. (1987), Performance Evaluation of Fork and Join Synchronization Primitives, *Acta Informatica*, **24**, 525-553.

Hariri, A. M. A. and Potts, C. N. (1997), A Branch-and-Bound Algorithm for the Two-Stage Assembly Scheduling Problem, *European Journal of Operational Research*, **103**, 547-556.

He, D. W. and Kusiak, A. (1996), Performance Analysis of Modular Products, *International Journal of Production Research*, **34**, 253-272.

Lee, C. Y., Cheng, T. C. E., and Lin, B. M. T. (1993), Minimizing the Makespan in the 3-Machine Assembly-type Flowshop Scheduling Problem, *Management Science*, **39**, 616-625.

Leung, J. Y. T., Li, H., and Pinedo, M. (2005a), Order Scheduling in an Environment with Dedicated Resources in Parallel, *Journal of Scheduling*, **8**, 355-386.

Leung, J. Y. T., Li, H., and Pinedo, M. (2005b), Order Scheduling Models: an Overview, *Multidisciplinary Scheduling*, **1**, 37-56.

Leung, J. Y. T., Li, H., and Pinedo, M. (2006a), Approximation Algorithms for Minimizing Total Weighted Completion Time of Orders on Identical Machines in Parallel, *Naval Research Logistics*, **53**, 243-260.

Leung, J. Y. T., Li, H., and Pinedo, M. (2006b), Scheduling Orders for Multiple Product Types with Due Date Related Objectives, *European Journal of Operational Research*, **168**, 370-389.

Norahari, Y. and Sundarrajan, P. (1995), Performability Analysis of Fork-Join Queueing Systems, *Journal of the Operational Research Society*, **46**, 1237-1249.

Potts, C. N., Sevast'janov, S. V., Strusevich, V. A., Wassenhove, L. N., and Zwaneveld, C. M. (1995), The Two-Stage Assembly Scheduling Problem : Complexity and Approximation, *Operations Research*, **43**, 346-355.

Sun, X., Morizawa, K., and Nagasawa, H. (2003), Powerful Heuristics to Minimize Makespan in Fixed, 3-Machine, Assembly-type Flowshop Scheduling, *European Journal of Operational Research*, **146**, 499-517.

Sung, C. S. and Yoon, S. H. (1998), Minimizing Total Weighted Completion Time at a Pre-assembly Stage Composed of Two Feeding Machines, *International Journal of Production Economics*, **54**, 247-255.

Tozkanan, A., Kirca, O., and Chung, C. S. (2003), A Branch-and-Bound Algorithm to Minimize the Total Weighted Flowtime for the Two-stage Assembly Scheduling Problem, *Computers*

- & Operations Research*, **30**, 309-320.
- Wang, G. and Cheng, T. C. E. (2007), Customer Order Scheduling to Minimize Total Weighted Completion Time, *to appear in Omega*.
- Yang, J. and Posner, M. E. (2005), Scheduling Parallel Machines for the Customer Order Problem, *Journal of Scheduling*, **8**, 49-74.
- Yoon, S. H. and Sung, C. S. (2005), Fixed Pre-assembly Scheduling on Multiple Fabrication Machines, *International Journal of Production Economics*, **96**, 109-118.