

아웃소싱 전략을 활용하는 두 단계 흐름생산라인에서 완료시간의 총합을 최소화하는 일정계획문제*

유재욱 · 이익선[†]
동아대학교 경영학과

Minimizing Total Completion Times in a Two-machine
Flowshop Scheduling with Outsourcing Strategy allowed

Jaewook Yoo · Ik Sun Lee[†]
School of Business, Dong-A University

■ Abstract ■

We treat a job scheduling in a 2-machine flow-shop problem with outsourcing strategy allowed. Jobs in the first machine are processed in-house or outsourced to the other companies. In this paper, all the considered jobs are determined to be in-house processed or outsourced. When a job is outsourced, then the firm should pay an outsourcing cost additionally. We want to minimize the sum of the outsourcing costs and the total completion times of finished jobs. In this paper, some solution properties are characterized, and then some heuristic algorithms and a branch-and-bound solution algorithm are derived. This paper evaluates finally the performance of the proposed algorithms during the numerical tests.

Keywords : Scheduling, Outsourcing, Two-machine Flowshop, Branch-and-Bound, Heuristic

논문접수일 : 2016년 01월 05일 논문게재확정일 : 2016년 02월 04일

논문수정일 : 2016년 02월 02일

* 이 논문은 동아대학교 학술연구비 지원에 의하여 연구되었음.

† 교신저자, lis1007@dau.ac.kr

1. 서론

아웃소싱(Outsourcing)은 기업이 수행해야 할 역할의 일부를 외부의 업체에게 위탁하는 것을 뜻한다. 기업의 경쟁력 강화를 위해서 아웃소싱은 이미 상당 부분 일반화된 전략으로 자리를 잡고 있으며 폭넓게 활용되고 있다. 본 논문은 효율적인 작업일정계획을 수립함에 있어서 아웃소싱을 활용하는 상황의 의사결정문제를 다룬다. 본 연구는 2단계의 생산설비를 가지는 제조상황 하에서, 첫 번째 설비의 가공을 직접적으로 처리할 수도 있지만, 이를 외부업체에 아웃소싱하는 옵션을 고려할 수도 있다. 첫 번째 설비에서 작업을 외부업체에 아웃소싱한다면 아웃소싱 비용을 추가적으로 지출해야 하며, 아웃소싱 리드타임이 경과한 후에 반제품의 작업을 인도받을 수 있다. 부품이 직접 가공되던지 아웃소싱 되던지에 무관하게 두 번째 설비의 가공은 회사 내에서 직접 처리하는 상황을 고려한다.

현재까지의 아웃소싱 연구들의 동향을 정리해보면 다음과 같다. 먼저 효과적인 아웃소싱 전략을 위한 시스템구조를 제안한 연구들이 있는데, Abdel-Malek et al.[1]은 다층 공급사슬에서 아웃소싱 전략을 구상하였으며, Kouvelis and Milner[7], Wu[19], Ngwenyama and Bryson[13]는 기업에서의 아웃소싱을 운영구조를 연구하였으며, Dekkers[4], Offodile and Abdel-Malek[14]는 경영관리기법들과 아웃소싱을 연계하는 방법론을 연구하였다.

Kim[6]은 보다 효과적인 아웃소싱 업체선정 방법론을 연구하였다. Yang et al.[20]은 아웃소싱 상황에서 재고 관리방안을 연구했으며, Huiskonen and Pirttila[5]는 아웃소싱 업체와 본사간의 조정이슈들을 논하였다. Tarakci et al.[17, 18]는 유지보수 분야에서의 아웃소싱을 고려하였다.

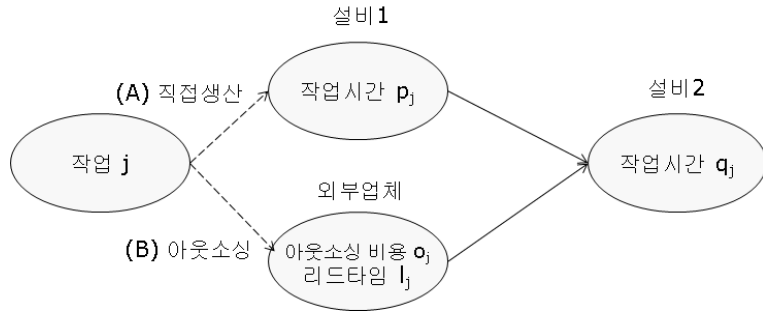
일정계획(Scheduling) 연구분야에서의 아웃소싱 전략은 Lee and Sung[8, 9]에 의해 최초 제안되었다. Lee and Sung[8, 9]는 단일설비에서 작업을 처리하는 상황에서 다양한 아웃소싱 상황들과 일정계획 비용들을 연구하였다. 이후에 Qi[15, 16], Choi and

Chung[3], Neto and Filho[12], Choi and Chung[3], Lee[10]은 두 단계의 흐름설비(two-machine flow-shop)에서 아웃소싱 전략을 고려하는 연구들을 수행하였는데, 이들 연구들은 작업들의 최종완료시간(makespan)과 관련된 일정계획을 수립연구들을 수행하였다.

본 연구는 두 단계 연속설비에서 완료시간의 총합과 아웃소싱비용의 합을 최소화하는 일정계획을 수립하는 것을 주요한 목적으로 삼는다. 두 단계 흐름설비에서 완료시간의 총합을 최소화하는 문제는 복잡도의 측면에서 매우 어려운 NP-complete 문제임이 이미 알려져 있다. 따라서 본 연구에서 다루는 문제도 당연히 NP-complete 문제임을 알 수 있으며, 이러한 사항을 고려하여 작은 크기의 문제상황에 대해서는 분지한계(branch-and-bound) 알고리즘을 제시하고, 또한 문제의 사이즈가 증가하는 상황에 대해서는 효율적으로 일정계획을 수립할 수 있는 휴리스틱 알고리즘을 제시하고자 한다.

2. 문제정의와 분석

본 논문은 두 설비가 연속으로 배치되어 있는 flowshop을 고려하고, 설비에서 처리해야 하는 n 개의 작업을 고려한다. 작업은 첫 설비에서 자체적으로 처리되거나, 외부회사에 아웃소싱될 수도 있지만, 대신에 두 번째 설비에서의 가공은 반드시 직접 처리해야 함을 가정한다. 첫 번째 설비에서 작업 i ($i=1, 2, \dots, n$)가 직접 처리된다면 가공시간 p_i 이 필요하고, 만일 i 가 외부로 아웃소싱된다면 아웃소싱 비용 o_i 을 지불해야 하며, 아웃소싱 리드타임 l_i 를 가정한다. 이 리드타임에는 외부업체로의 주문시간, 가공시간, 배송시간 등이 포함된다. 만일 작업 i 가 아웃소싱된다면, 첫 번째 설비에서의 처리시간은 고려할 필요가 없게 되며, 두 번째 설비에서의 가공은 리드타임 l_i 이후에 시작할 수 있게 된다. 두 번째 설비에서는 아웃소싱을 고려하지 않으며, 두 번째 설비에서 작업의 처리시간은 q_i 으로 표현한다. 본 연구는 작업들의 가공순서를 순열일정(permutation schedule)



<그림 1> 본 연구가 고려하는 문제상황

을 따르는 것으로 가정한다. 본 연구가 고려하는 문제 상황을 표현하면 다음 <그림 1>과 같다.

설비 1에서 직접적으로 처리하는 작업들의 집합을 $\bar{\pi}$ 이라 표현하고, 아웃소싱되는 작업들의 집합을 π 이라고 표현할 때, 아웃소싱에 투입되는 비용 $OC(\pi)$ 는 $\sum_{i \in \pi} o_i$ 으로 표현할 수 있다. 본 논문은 작업 완료시간의 총합과 아웃소싱 비용의 합산을 최소화하는 작업일정계획을 수립하는 것으로 목적으로 한다. 본 논문에서의 총비용 TC 는 작업완료시간의 총합과 아웃소싱 비용의 합산으로서 다음과 같이 표현된다.

$$TC = \delta \sum_{i \in \pi} o_i + (1 - \delta) \sum_{i \in \pi} C_i \quad (1)$$

C_i 는 작업 i 의 설비 2에서의 작업완료시간을 의미하고, δ 는 작업의 완료시간과 아웃소싱 비용과의 중요도를 반영하기 위해 0과 1사이 실수 값으로 상정한다. 본 논문에서 다루는 의사결정문제를 수리적 모형으로 수립한다면 다음과 같다.

[본 연구의 수리적 모형]

$$\text{Min} \quad \delta \sum_i o_i y_i + (1 - \delta) \sum_i C_i \quad (2)$$

$$\text{s.t.} \quad C_i \geq x_{ik} \sum_{u=1}^k \sum_{j=1}^n p_j x_{ju} + l_j y_j + q_i \quad \text{for } \forall i, k \quad (3)$$

$$C_i \geq x_{ik} \sum_j x_{jk-1} C_j + q_i \quad \text{for } \forall i, k \quad (4)$$

$$\sum_k x_{ik} + y_i = 1, \quad \text{for } \forall i \quad (5)$$

$$x_{ij}, y_i \in \{0, 1\}.$$

수리적 모형의 변수들을 설명하면 다음과 같다. y_i 변수는 작업 i 가 아웃소싱되는 경우에 상수 1을 가지고, 직접 처리되는 경우에는 0을 가진다. x_{ik} 변수는 작업 i 가 직접 처리되는 경우인데 k 번째 위치에서 처리된다면 1을 가지고, 나머지의 경우에는 0을 가진다. 제약식 (2)는 총비용을 뜻하는 목적함수이며, 식 (3), 식 (4)는 각 작업별 처리완료시간을 구하기 위한 계산식들이다. 제약식 (5)는 작업 i 가 아웃소싱되거나 직접 처리됨을 의미한다.

수리적 모형에서 제약식 (4), 제약식 (5)는 결정변수들의 곱을 포함하고 있기 때문에, 이러한 수리적 모형에서 솔루션을 효과적으로 찾는 것은 쉬운 일이 아니다. 더욱이 본 연구에서 다루는 의사결정문제는 이미 잘 알려진 NP-complete 문제의 성질을 포함하고 있음을 이미 논의하였다. 따라서 본 연구는 효율적인 휴리스틱을 제시하는 것을 주요 목적으로 삼는다. 이제부터는 효율적인 휴리스틱을 개발하기 위한 방안을 제시하고자 한다.

먼저 직접적으로 처리되는 작업들의 처리순서가 이미 결정되어 있는 상황을 가정해보자. 아웃소싱 작업들은 설비 1에서의 처리에서는 제외되며, 외부 업체로부터 리드타임 l_i 이후에 배송되어진다. 설비 1에서 직접적으로 처리되는 작업들의 순서가 이미 결정되어 있는 상황을 가정한다면, 앞에서 제시된 수리적 모형의 형태가 보다 간소화될 수 있으며, 새로운 수리적 모형은 다음과 같이 제시될 수 있다.

[새로운 수리적 모형]

$$\begin{aligned} \text{Min} \quad & \delta \sum_i o_i y_i + (1-\delta) \sum_i C_i \\ \text{s.t.} \quad & C_i \geq \sum_{j=1}^i p_j x_j + l_j y_j + q_i \quad \text{for } \forall i \quad (7) \\ & C_i \geq C_{i-1} + q_i \quad \text{for } \forall i \quad (8) \\ & x_i + y_i = 1, \quad \text{for } \forall i \quad (9) \\ & x_i, y_i \in \{0, 1\}. \end{aligned}$$

x_i 는 i 번째에 위치한 작업이 설비에서 직접 처리된다면 1을 가지고, 아니면 0을 가진다. 나머지는 기존의 수학적 모형에서의 변수들을 동일하게 활용한다.

여기서는 아웃소싱 대상에서 원천적으로 배제하는 것이 이득인 작업들을 선별하는 방안을 고려한다. 아래의 Property 1을 만족하는 작업은 아웃소싱 대상에서 원천적으로 배제하는 것이 보다 이득이다.

Property 1. 다음 조건을 만족하는 작업 j 이라면, 직접 처리하는 것이 보다 이득이다.

$$\delta o_j > (1-\delta) \max \left\{ \sum_i (p_j + q_j), n(p_j + q_j) \right\} \quad (10)$$

Proof. 만일 작업 j 가 아웃소싱되면 추가적으로 지불하게 되는 비용은 최소 δo_j 이다. 대신에 작업 j 를 직접적으로 처리한다면 비용의 추가 금액은 최대 $(1-\delta) \max \left\{ \sum_i (p_j + q_j), n(p_j + q_j) \right\}$ 이다. 그러므로 만일 식 (10)이 성립하면 작업 j 를 직접 처리하는 것이 보다 이득임을 쉽게 알 수 있다. \square

3. 휴리스틱 알고리즘들

본 장에서는 일정계획문제에 대해서 동적계획 및 휴리스틱 알고리즘 등을 제안한다.

3.1 동적계획 알고리즘

본 연구에서 고려하는 문제가 NP-complete 문제

라는 점을 이미 논의하였다. 이번 절에서는 의사결정문제 자체가 아니라 변화하는 조건하에서 효율적인 해법들을 논의하고자 한다. 먼저 본 연구의 의사결정문제가 NP-complete 문제지만, 직접적으로 처리하는 작업들의 처리순서가 미리 결정되어 있는 경우에 활용할 수 있는 알고리즘을 제안하고 하는데, 다음의 동적계획 알고리즘을 활용할 수 있다.

<동적계획 알고리즘>

정렬 : 미리 결정되어진 처리순서대로 작업들을 정렬한다.

가치함수 $f_j(x_1, y_1, y_2)$: 1부터 j 까지 작업들이 $[0, x_1]$ 동안 설비 1에서 가공되고, $[y_1, y_2]$ 동안 설비 2에서 가공되는 상황에서 목적함수의 최소값이다.

초기조건 : $f_0(0, 0, 0) = 0$

최적해 : $\min_{((x_1, y_1, y_2) | 0 \leq x_1 \leq A, 0 \leq y_1 \leq y_2 \leq B)} f_n(x_1, y_1, y_2)$,

$$\text{where } A = \sum_{j=1}^n p_j, B = \sum_{j=1}^n (p_j + q_j)$$

순환식 :

$$f_j(x_1, y_1, y_2) = \begin{cases} \infty, & \text{if } y_1 > y_2 - q_j & (i) \\ \min_{0 \leq k \leq (y_2 - q_j - x_1)} \{ f_{j-1}(x_1 - p_j, y_1, y_2 - q_j - k) + (1-\delta)y_2 \} & (ii) \\ \infty, & \text{if } x_1 \leq y_2 - q_j & (iii) \\ f_{j-1}(x_1, y_1, y_2 - q_j) + (1-\delta)y_2 + \delta o_j & (iv) \end{cases}$$

동적계획 알고리즘의 순환식 (i)는 설비 2에서 작업 j 를 직접 처리할 여유가 없기 때문에 무한대의 값을 할당한다. 순환식 (ii)는 작업 j 를 직접 처리하는 경우의 최소비용을 뜻한다. 순환식 (iii)는 작업 j 가 아웃소싱되지만 리드타임이 촉박하여 제대로 처리할 수 없기 때문에 무한대의 비용을 대입한다. 마지막 순환식 (iv)는 작업 j 를 원활히 아웃소싱 처리하는 경우의 최소비용을 뜻한다. 본 연구에서 제안하는 동적계획 알고리즘의 상태(state)의 수를 계산해보면 최대 AB^2 으로서 결과적으로 $O(AB^2)$ 시간복잡도를 가진다. 즉 시간복잡도가 크므로 본 동적계획 알고리즘은 그다지 실용적이라고 말할 수는 없다.

따라서 다음 장에서는 보다 효율적인 휴리스틱 알고리즘들을 제시한다.

3.2 휴리스틱 알고리즘

본 절에는 짧은 시간 내에 효율적인 일정계획을 수립할 수 있는 휴리스틱 알고리즘을 제시하고자 한다. 본 절에서는 모두 5개의 휴리스틱들을 제시한다. 5개의 휴리스틱들은 모두 Greedy 유형의 알고리즘들이다. Greedy 유형의 알고리즘이란 현재 상태에서 이득이 클 것 같은 작업을 일단 우선적으로 처리하는 방식을 매번 반복하는 것이다. 이러한 Greedy 유형의 알고리즘은 평균적으로는 성능이 우수한 것으로 알려져 있지만, 최적해를 보장할 수는 없다는 한계점을 지니고 있다.

본 절에서 제시되는 휴리스틱들은 제 3.1절에서 다룬 개념과 유사하게 작업들의 처리순서를 먼저 결정해준다. 이러한 처리순서가 최적이라고 말할 수는 없지만 문제의 복잡도를 줄여준다는 측면에서는 매우 활용성이 높다고 말할 수 있다. 첫 번째 휴리스틱은 두 설비에서 직접 처리하는 작업시간의 합을 오름차순으로 정렬해놓고, 이러한 순서 속에서 아웃소싱의 이득이 가장 큰 작업을 선별해나가는 방식을 취하고 있다. 세부적인 절차는 다음과 같다.

<휴리스틱 H1>

단계 0 : $(p_j + q_j)$ 의 오름차순으로 순서를 결정하고 작업들을 정렬한다.

단계 1 : 모든 작업들을 직접 가공한다는 가정하에서 완료시간의 총합 TC 를 계산한다.

직접 처리하는 작업들의 집합 $\pi = \{1, 2, \dots, n\}$, 아웃소싱 집합 $\tilde{\pi} = \emptyset$ 으로 초기화하고, 아웃소싱비용 $OC = 0$ 으로 초기화한다.

단계 2 : π 에 속하는 작업 j 를 아웃소싱하는 상황을 가정하자.

작업 j 가 아웃소싱된다면 작업 j 는 설비 2에서 아웃소싱 리드타임 l_j 이후에 처리될 수 있다.

변경된 일정계획에 대해서 새로운 완료시간의 총합 \widetilde{TC}_j 를 계산한다.

단계 3 : 모든 작업 $j \in \pi$ 에 대해서 $\widetilde{TC}_j \geq TC$ 이라면 $\tilde{\pi}$ 에 속하는 작업들은 아웃소싱하고, π 에 속하는 작업들은 직접 처리하는 것으로 결론내리고, 알고리즘을 종료한다.

단계 4 : $\widetilde{TC}_j < TC$ 을 만족하는 작업들 중에서, 최소 \widetilde{TC}_j 값을 가지는 작업 j 를 선정한다.

단계 5 : 작업 j 를 아웃소싱한다. j 를 집합 π 에서 제거하고, 집합 $\tilde{\pi}$ 에 포함시킨다.

TC 값을 \widetilde{TC}_j 으로 업데이트하고, $OC = OC + o_j$ 으로 업데이트한다. 단계 2로 이동한다.

두 번째 휴리스틱은 첫 번째 설비에서의 처리시간 (p_j)에 따라서 작업들을 오름차순으로 정렬해놓고, 이러한 순서 속에서 아웃소싱의 이득이 가장 큰 작업을 선별해나가는 방식을 취한다. 세 번째 휴리스틱은 두 번째 설비에서의 처리시간 (q_j)에 기반하여 작업들을 오름차순으로 정렬해놓고, 이러한 순서 속에서 아웃소싱의 이득이 가장 큰 작업을 선별해나가는 방식을 취한다. 세부적인 절차는 다음과 같다.

<휴리스틱 H2>

단계 0 : (p_j) 의 오름차순으로 순서를 결정하고 작업들을 정렬한다.

단계 1~5 : 휴리스틱 H1의 내용과 동일하다.

<휴리스틱 H3>

단계 0 : (q_j) 의 오름차순으로 순서를 결정하고 작업들을 정렬한다.

단계 1~5 : 휴리스틱 H1의 내용과 동일하다.

네 번째 휴리스틱 H4는 숫자 비율 $\left(\frac{p_j}{q_j}\right)$ 에 따라서 작업들을 오름차순으로 정렬해놓고, 이러한 순서 속에서 아웃소싱의 이득이 가장 큰 작업을 선별해나가는

는 방식을 취한다. 다섯 번째 휴리스틱 H5는 숫자 비율 $\left(\frac{p_j}{q_j \rho_j}\right)$ 에 기반하여 작업들을 오름차순으로 정렬해놓고, 이러한 순서 속에서 아웃소싱의 이득이 가장 큰 작업을 선별해나가는 방식을 취한다. 세부적인 절차는 다음과 같다.

<휴리스틱 H4>

단계 0 : $\left(\frac{p_j}{q_j}\right)$ 의 오름차순으로 순서를 결정하고 작업들을 정렬한다.

단계 1~5 : 휴리스틱 H1의 내용과 동일하다.

<휴리스틱 H5>

단계 0 : $\left(\frac{p_j}{q_j \rho_j}\right)$ 의 오름차순으로 순서를 결정하고 작업들을 정렬한다.

단계 1~5 : 휴리스틱 H1의 내용과 동일하다.

4. 분지한계법

4.1 분지규칙

본 절에서는 최적해를 구하는 분지한계법(Branch-and-bound method)를 제시하고자 한다. 본 연구의 분지한계법은 하나의 노드를 임의로 선정하고, 아직 정해지지 않은 작업을 선택하여 해당 작업을 직접적으로 처리하는 노드를 새롭게 생성하고, 또 선택된 작업을 아웃소싱하는 경우에 해당하는 노드를 추가하는 방법을 반복적으로 활용해나가는 방식을 취한다. 본 논문의 분지한계법은 깊이우선검색(depth-first search)을 활용하여 노드를 선택한다. 다시 말하면 가장 많은 작업들을 포함하고 있는 노드를 먼저 선정함을 의미한다.

4.2 한계규칙(Bounding Rule)

본 논문의 분지한계법에서 활용하는 초기상한 값

으로 제 3장에서 제시된 휴리스틱 알고리즘들의 값을 활용한다. 5개 휴리스틱들이 제시하는 값들 중에서 가장 뛰어난 해를 초기상한 값으로 활용한다. 불필요한 노드를 제거하는 방식으로 하한값을 활용하는데, 본 연구에서 활용하는 하한값 LB_r 은 다음과 같이 계산된다.

$$LB_r = TC(\pi_r) + (1-\delta)\{K \times |\tilde{\pi}_r| + SPT_{\pi_r}^-(q_j)\},$$

$|\tilde{\pi}_r|$ 는 집합 $\tilde{\pi}_r$ 에 속한 작업들의 수를 뜻하며, K 는 집합 π_r 의 마지막 작업의 설비 2에서의 작업완료시점을 뜻한다. 또한 $SPT_{\pi_r}^-(q_j)$ 는 집합 $\tilde{\pi}_r$ 에 속한 작업들을 q_j 의 SPT순으로 정렬하여 얻은 완료시간의 총합을 뜻한다.

5. 성능 평가

본 연구에서 제시된 휴리스틱 알고리즘들과 분지한계법의 성능을 분석하고자 한다. 알고리즘의 성능을 분석하기 위한 파라미터 설정을 소개하고자 한다. 먼저 설비 1과 설비 2에서의 처리시간 p_i 와 q_i 는 $U(1, 20)$ 에서 생성시킨다. 기호 $U(a, b)$ 는 이산형 균등분포를 뜻한다. 아웃소싱비용 o_i 는 $U(10, 50)$ 를 활용하여 생성시킨다. 목적식에서 활용되는 상수 δ 는 $U(0.2, 0.8)$ 으로부터 임의의 생성한다. 성능평가에 활용되는 작업들의 수는 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24의 총 11가지를 활용한다. 각 작업 수 별로 20개씩의 인스턴스들을 임의로 생성하였다.

성능테스트 결과는 <표 1>에서 정리되었는데 분지한계법에 대하여는 최소비용의 일정을 구하기까지의 노드의 평균수, 최대수, 수행시간의 평균, 최대를 제시하였다. NO는 20개의 문제들 중에서 분지한계법이 최적해를 찾아낸 문제들의 수를 알려준다. NO는 최대 20까지 가질 수 있음을 확인할 수 있다. 표를 확인해보면 분지한계법은 비교적 적지 않은 수의 작업들에 대해서 최적해를 제시하고 있음을 알 수 있다.

제 3장에서 제시된 휴리스틱들의 성능을 평가해 보고자 한다. 휴리스틱을 통해 구한 해들과 분지한계법을 통해 얻은 최적해와의 Gap(%)의 평균치와 최대치를 <표 2>에 제시하였다. 표에서 활용하는 Gap(%)의 계산은 다음의 식과 같다. $Gap(\%) = \frac{(Heuristic - OPT)}{OPT} \times 100$, 단 OPT가 분지한계법을 활용하여 얻은 최적해이고 Heuristic은 각 휴리스틱들로부터 얻은 해를 뜻한다. <표 2>의 결과를 확인

<표 1> B&B 알고리즘 성능 요약

작업 수	노드 수		계산시간		NO
	평균	최대	평균	최대	
4	14	37	0.00	0.00	20
6	132	297	0.00	0.01	20
8	837	8528	0.00	0.03	20
10	3145	16461	0.01	0.06	20
12	52107	310416	0.25	1.44	20
14	204963	1555956	1.34	9.68	20
16	2293389	23302257	19.08	191.80	20
18	8329429	121865775	95.66	1351.53	19
20	14239095	160399089	242.04	2691.81	16
22	18161911	178318107	401.22	3863.25	14
24	38240688	216776178	1081.48	6031.44	12

해보면 휴리스틱 H1, H2가 H3, H4, H5에 비해 월등한 성능을 보여줌을 알 수 있다. 따라서 이후는 휴리스틱 H1, H2에 대해서만 성능을 분석하도록 한다.

상수값 δ 의 추이에 따른 휴리스틱의 성능을 평가하여 그 결과를 <표 3>에 제시하였다. δ 값은 $U(0.3, 0.4)$, $U(0.4, 0.5)$, $U(0.5, 0.6)$ 을 상정하였다. δ 값이 증가할수록 휴리스틱 알고리즘들의 성능이 전반적으로 향상되는 경향을 관찰할 수 있다.

<표 3> 상수 δ 값의 변동에 따른 휴리스틱 성능평가

δ	n	H1		H2	
		Aver. Gap(%)	Max. Gap(%)	Aver. Gap(%)	Max. Gap(%)
[0.3, 0.4]	12	17.01	34.74	19.00	32.23
	14	17.71	40.60	22.48	38.41
	16	17.65	35.13	24.45	40.19
[0.4, 0.5]	12	14.13	26.35	16.53	25.99
	14	16.18	34.45	20.38	34.25
	16	14.83	35.93	23.54	41.21
[0.5, 0.6]	12	13.58	31.22	17.33	32.67
	14	14.51	29.30	19.39	46.67
	16	17.12	27.93	21.19	32.64

<표 2> 휴리스틱들에 대한 성능테스트

	H1		H2		H3		H4		H5	
	Aver. Gap(%)	Max. Gap(%)	Aver. Gap(%)	Max. Gap(%)	Aver. Gap(%)	Max. Gap(%)	Aver. Gap(%)	Max. Gap(%)	Aver. Gap(%)	Max. Gap(%)
4	7.20	38.26	6.96	20.77	23.87	74.75	11.66	24.00	12.32	24.00
6	6.50	20.81	9.03	33.65	22.13	59.71	21.91	56.44	25.19	95.71
8	12.81	29.93	12.15	23.15	31.61	76.28	27.91	74.25	26.93	52.82
10	14.20	36.03	18.77	35.12	33.59	60.79	35.79	60.54	34.73	59.08
12	15.67	35.14	16.78	29.83	35.41	61.95	35.46	53.69	35.16	55.78
14	13.00	35.35	16.76	36.29	31.12	75.48	38.98	62.10	39.36	63.85
16	22.02	200.75	28.07	192.00	49.01	253.86	44.75	220.53	44.03	221.16
18	18.51	43.06	23.78	39.34	38.90	65.28	45.34	61.80	43.13	60.40
20	16.24	30.76	23.93	43.20	41.91	70.51	46.41	63.39	45.30	65.76
22	15.16	36.11	25.70	40.74	38.54	57.46	48.44	58.31	46.07	62.39
24	19.35	36.26	27.15	37.55	41.22	65.29	48.85	67.95	46.60	70.07

아웃소싱 비용 o_j 의 추이에 따른 휴리스틱들의 성능을 평가하여 그 결과를 <표 4>에 제시하였다. o_j 값은 $U(10, 20)$, $U(20, 30)$, $U(30, 40)$ 을 상정하였으며, 아웃소싱 비용이 증가할수록 휴리스틱 알고리즘들의 성능이 전반적으로 향상되는 경향을 관찰할 수 있다.

<표 4> 아웃소싱 비용의 p_j 변동에 따른 휴리스틱 성능평가

o_j	n	H1		H2	
		Aver. Gap(%)	Max. Gap(%)	Aver. Gap(%)	Max. Gap(%)
[10, 20]	12	16.30	27.25	20.29	35.16
	14	18.01	40.03	22.87	41.21
	16	21.32	40.65	26.18	40.62
[20, 30]	12	12.77	30.43	18.21	26.49
	14	14.86	49.65	20.99	41.54
	16	16.89	44.73	21.81	40.58
[30, 40]	12	13.64	35.35	15.25	30.00
	14	13.08	40.20	18.24	28.75
	16	13.77	28.73	17.58	25.90

설비 1에서의 처리시간 p_j 의 추이에 따른 휴리스틱들의 성능을 평가하여 그 결과를 <표 5>에 제시하였다. p_j 값은 $U(1, 20)$, $U(20, 40)$, $U(40, 60)$ 을 상정하였으며, 처리시간 p_j 가 증가할수록 휴리스틱 알고리즘들의 성능이 전반적으로 하락하는 경향을 관찰할 수 있다.

<표 5> 첫 번째 설비에서의 처리시간 변동에 따른 휴리스틱 성능평가

p_j	n	H1		H2	
		Aver. Gap(%)	Max. Gap(%)	Aver. Gap(%)	Max. Gap(%)
[1, 20]	12	18.31	39.67	18.41	32.33
	14	14.68	31.32	20.23	42.48
	16	17.68	34.34	22.20	40.38
[20, 40]	12	45.29	72.99	40.65	65.69
	14	46.26	60.25	42.39	56.00
	16	53.73	77.98	48.88	73.33
[40, 60]	12	51.81	86.37	49.13	81.14
	14	62.28	95.78	59.61	94.24
	16	68.49	81.12	65.76	79.13

설비 2에서의 처리시간 q_j 의 추이에 따른 휴리스틱들의 성능을 평가하여 그 결과를 <표 6>에 제시하였다. q_j 값은 $U(1, 20)$, $U(20, 40)$, $U(40, 60)$ 을 상정하였으며, 처리시간 q_j 가 증가할수록 휴리스틱 알고리즘들의 성능이 전반적으로 향상됨을 관찰할 수 있다.

<표 6> 두 번째 설비에서의 처리시간 q_j 변동에 따른 휴리스틱 성능평가

q_j	n	H1		H2	
		Aver. Gap(%)	Max. Gap(%)	Aver. Gap(%)	Max. Gap(%)
[1, 20]	12	16.90	33.62	19.68	33.74
	14	16.89	35.60	21.19	31.45
	16	15.68	35.25	20.11	37.85
[20, 40]	12	3.77	8.42	10.31	17.35
	14	3.87	6.57	10.71	18.36
	16	3.93	8.39	11.32	15.28
[40, 60]	12	1.85	2.82	5.55	9.49
	14	1.80	2.81	5.80	9.34
	16	2.05	4.21	6.25	9.78

6. 결 론

본 연구는 두 생산설비에서 가공하여 작업을 처리하는 상황에서 아웃소싱을 활용하는 일정계획문제를 다루고 있다. 본 연구에서 고려하는 아웃소싱은 설비 1에서 작업의 처리를 직접적으로 수행하거나 외부업체에 위탁하는 옵션을 선택할 수 있음을 뜻한다. 이러한 상황에서 작업완료시간의 총합과 아웃소싱 비용의 합산을 최소화하는 일정계획을 수립하기 위하여 동적계획법, 분지한계법, 휴리스틱 알고리즘 등을 제안하였다.

본 논문은 시스템에서 시간적인 압박이 커서 아웃소싱 전략을 활용하는 환경에서 효율적으로 적용될 수 있을 것이다. 기존의 일정계획 연구에 비해서 본 논문은 아웃소싱을 접목하는 새로운 경쟁환경의 의사결정문제를 다루고 있으므로 보다 가변적인 상황과 제약들을 고려하는 추가적인 연구들을 검토할 수 있을 것이다.

참 고 문 헌

- [1] Abdel-Malek, L., T. Kullpattaranirun, and S. Nanthavanij, "A Framework for Comparing Outsourcing Strategies in Multi-layered Supply Chains," *International Journal of Production Economics*, Vol.97, No.3(2005), pp.318-328.
- [2] Choi, B.C. and J. Chung, "Two-machine flow shop scheduling problem with an outsourcing option," *European Journal of Operational Research*, Vol.213, No.1(2011), pp.66-72.
- [3] Chung, D.Y. and B.C. Choi, "Outsourcing and scheduling for two-machine ordered flow shop scheduling problems," *European Journal of Operational Research*, Vol.226, No.1(2013), pp.46-52.
- [4] Dekkers, R., "Decision Models for Outsourcing and Core Competencies in Manufacturing," *International Journal of Production Research*, Vol.38, No.17(2000), pp.4085-4096.
- [5] Huiskonen, J. and T. Pirttila, "Lateral Coordination in a Logistics Outsourcing Relationship," *International Journal of Production Economics*, Vol.78, No.2(2002), pp.177-185.
- [6] Kim, B., "Dynamic Outsourcing to Contract Manufacturers with Different Capabilities of Reducing the Supply Cost," *International Journal of Production Economics*, Vol.86, No.1(2003), pp.63-80.
- [7] Kouvelis, P. and J.M. Milner, "Supply Chain Capacity and Outsourcing Decisions : the Dynamic Interplay of Demand and Supply Uncertainty," *IIE Transactions*, Vol.34, No.8(2002), pp.717-728.
- [8] Lee, I.S. and C.S. Sung, "Minimizing Due Date Related Measures for a Single Machine Scheduling with Outsourcing Allowed," *European Journal of Operational Research*, Vol.186, No.3(2008a), pp.931-952.
- [9] Lee, I.S. and C.S. Sung, "Single Machine Scheduling with Outsourcing Allowed," *International Journal of Production Economics*, Vol.111, No.2(2008b), pp.623-634.
- [10] Lee, I.S., "A Two-machine Flowshop Scheduling with Outsourcing Strategy allowed," *Society of Korea Industrial and Systems Engineering*, Vol.37, No.3(2014), pp.1113-121.
- [11] Lee, K. and B.C. Choi, "Two-stage production scheduling with an outsourcing option," *European Journal of Operational Research*, Vol.213, No.3(2011), pp.489-497.
- [12] Neto, R.F.T. and M.G. Filho, "An ant colony optimization approach to a permutational flowshop scheduling problem with outsourcing allowed," *Computers and Operations Research*, Vol.38, No.9(2011), pp.1286-1293.
- [13] Ngwenyama, O.K. and N. Bryson, "Making the Information Systems Outsourcing Decision : A Transaction Cost Approach to Analyzing Outsourcing Decision Problems," *European Journal of Operational Research*, Vol.115, No.2(1999), pp.351-367.
- [14] Offodile, O.F. and L.L. Abdel-Malek, "The Virtual Manufacturing Paradigm : The Impact of IT/IS Outsourcing on Manufacturing Strategy," *International Journal of Production Economics*, Vol.75, No.1-2(2002), pp.147-159.
- [15] Qi, X., "Outsourcing and production scheduling for a two-stage flowshop," *International Journal of Production Economics*, Vol.129, No.1(2011), pp.43-50.
- [16] Qi, X., "Two-stage production scheduling with an option of outsourcing from a remote supplier," *Journal of System Science and System Engineering*, Vol.18, No.1(2009), pp.1-15.
- [17] Tarakci, H., K. Tang, H. Moskowitz, and R.

- Plante, "Incentive Maintenance Outsourcing Contracts for Channel Coordination and Improvement," *IIE Transactions*, Vol.38, No.8(2006a), pp.671-684.
- [18] Tarakci, H., K. Tang, H. Moskowitz, and R. Plante, "Maintenance Outsourcing of a Multi-process Manufacturing System with Multiple Contractors," *IIE Transactions*, Vol.38, No.1 (2006b), pp.67-78.
- [19] Wu, F., H.Z. Li, L.K. Chu, and D. Sculli, "An Outsourcing Decision Model for Sustaining Long-term Performance," *International Journal of Production Research*, Vol.43, No.12 (2005), pp.2513-2535.
- [20] Yang, J., X. Qi, and Y. Xia, "A Production-inventory System with Markovian Capacity and Outsourcing Option," *Operations Research*, Vol.53, No.2(2005), pp.328-349.