

아웃소싱을 고려한 병렬기계 일정계획 연구

이익선* · 윤상흠**†

*동아대학교 경영학부

**영남대학교 경영학부

A Parallel Machine Scheduling Problem with Outsourcing Options

Ik Sun Lee* · Sang Hum Yoon**†

*School of Business, Dong-A University

**School of Management, Yeungnam University

This paper considers an integrated decision for scheduling and outsourcing(or, subcontracting) of a finite number of jobs(or, orders) in a time-sensitive make-to-order manufacturing environment. The jobs can be either processed in a parallel in-house facilities or outsourced to subcontractors. We should determine which jobs should be processed in-house and which jobs should be outsourced. And, we should determine the schedule for the jobs to be processed in-house. If a job is determined to be processed in-house, then the scheduling cost(the completion time of the job) is imposed. Otherwise(if the job should be outsourced), then an additional outsourcing cost is imposed. The objective is to minimize the linear combination of scheduling and outsourcing costs under a budget constraint for the total available outsourcing cost. In the problem analysis, we first characterize some solution properties and then derive dynamic programming and branch-and-bound algorithms. An efficient heuristic is also developed. The performances of the proposed algorithms are evaluated through various numerical experiments.

Keywords : Scheduling, Outsourcing, Parallel Machine, Branch-and-Bound, Heuristic

1. 서 론

아웃소싱(Outsourcing : 외부위탁)이란 기업의 특정업무나 기능을 전문성을 가진 외부업체에 위탁하는 것을 의미하며, 기업은 아웃소싱을 통해 업무수행의 리드타임을 줄이고 비용을 절감함으로써 시간과 비용측면의 경쟁력을 강화하게 된다.

아웃소싱의 최근 추세는 단위기능의 위탁에서부터 보다 포괄적인 업무 프로세스와 기능을 대상으로 확대되

고 있으며, 연구개발과 같은 기업의 핵심기능도 필요에 따라 아웃소싱의 대상이 되고 있다. 이러한 아웃소싱의 확대는 기업의 자체 프로세스역량의 향상을 저해하고 회사의 기능이 외부업체에 맡겨지는 과정에서 여러 위험요인이 존재하는 것도 사실이지만, 설비, 인력 등의 고정비용의 투자로 인한 재무적 부담을 덜 수 있을 뿐 아니라 제품과 시장의 생명주기(life cycle)가 급속히 짧아지고 결국 시간이 기업간 주요 경쟁요인이 되고 있는 현실에서 기업 경쟁력강화를 위한 필수불가결한 선택이

논문접수일 : 2008년 08월 21일 논문수정일 : 2008년 09월 03일 게재확정일 2008년 09월 03일

† 교신저자 shyoon@ynu.ac.kr

※ 이 논문은 동아대학교 학술연구비 지원에 의하여 연구되었음.

라고 할 수 있다. 특히, 공급자와 생산자로 연결되는 공급사슬망(SCM)에서 각종 변동성의 위험요인으로부터 생산시스템을 보다 강건(robust)하고 유연(flexible)하게 유지하기 위해서 공급자에 대한 다양한 아웃소싱 수단을 확보하고 활용하는 것은 생산자입장에서 매우 중요한 생산전략중 하나이다.

본 연구는 제조업의 생산기능에서 이러한 아웃소싱수단이 존재하는 상황에서의 생산일정계획문제를 고려하고 있다. 즉, 개별 작업(또는 주문)의 처리가 자체생산 설비를 통해 직접 수행되거나 외부설비(아웃소싱)에 위탁되어 조달되는 두 가지 선택이 모두 존재하는 경우에 수행해야 할 전체 작업에서 아웃소싱될 작업들에 대한 선택(job selection)과 자체 수행될 작업들에 대한 일정계획(job scheduling)을 동시에 결정하는 의사결정 문제를 고려하고 있다. 자체생산설비는 병렬기계숍(parallel machine shop)을 고려하였으며 어떤 작업이 자체설비를 통해 처리될 경우에는 해당 작업에 대한 일정계획비용(scheduling cost)이 발생하게 되고 이를 해당 작업의 완료시간(completion time)으로 모델링 하였다. 또한, 그 작업이 아웃소싱되는 것으로 결정될 경우에는 추가적인 아웃소싱 비용이 발생하게 되며, 그 작업이 외부로부터 생산되어 조달완료되는 시점이 해당 작업의 완료시간이 된다. 본 연구에서는 이러한 아웃소싱으로 인한 총 아웃소싱비용에 일종의 예산제약(budget constraint)이 존재하는 경우에 총 일정계획비용과 총 아웃소싱비용을 합한 전체 비용이 최소화 될 수 있는 아웃소싱 작업의 선택과 자체생산 작업에 대한 일정계획을 동시에 결정하게 된다.

아웃소싱과 관련한 기존 연구들을 살펴보면 다음과 같다. 먼저, 아웃소싱 전략의 성과를 분석한 연구들이 수행되었다. 즉, 시스템 운영 측면에서, 아웃소싱 전략이 얼마만큼의 비용절감 및 이득을 가져다 줄 것인가를 정량적으로 분석한 것으로써 Nicholson et al.[12], Kaipia and Tanskanen[6], Cachon and Harker[3] 등의 연구가 있다.

둘째로, 아웃소싱 전략을 원활하게 관리하기 위한 구조/framework를 제안하는 연구들이 수행되었다. 예를 들어, Abdel-Malek et al.[1]은 다층구조 공급사슬(multi-layered supply chains)에서 아웃소싱 전략의 성능을 평가하기 위한 구조를 제안하였다. 또한, Wu et al.[16], Kouvelis and Milner[8], Ngwenyama and Bryson[11]은 기업의 아웃소싱 관련 업무를 효율적으로 관리하기 위한 의사결정 구조를 제안한 바 있으며, Offodile and Abdel-Malek[13], Dekkers[4]는 아웃소싱과 기존의 경영관리기법들을 연계하기 위한 의사결정구조를 제안하였다.

마지막으로, 아웃소싱의 효율을 극대화하기 위한 타 생산운영관리 기능들과의 접목에 관한 연구들도 활발히

진행되고 있다. Kim[7]은 아웃소싱을 담당할 공급자 선정과 관련한 효율적 방안에 대해서 연구하였다. Yang et al.[17]은 아웃소싱 상황 하에서 재고 운용 방안을 연구하였으며, Huiskonen and Pirttila[5]는 아웃소싱 제조회사와 고객회사간의 조직간 조정(coordination)과 관련한 문제들에 대해 논의하였다. Tarakci et al.[14, 15]은 유지 및 보수(maintenance activities)활동에서의 아웃소싱 전략에 대해서 연구하였다.

본 연구는 위의 관련연구고찰에서 마지막 범주에 해당하며, 전통적인 생산일정계획분야와 아웃소싱수단의 효율적 접목을 고려하고 있다. 기존의 생산일정계획 연구의 대부분은 모든 작업을 자체 처리하는 상황만을 고려하여 다양한 생산시스템에서의 일정계획을 결정하는 문제들을 고려해왔다. 이에 반해 본 연구에서는 아웃소싱으로 인한 추가적인 비용요인과 자체생산으로 인한 일정계획비용을 포함한 총 비용을 최소화 할 수 있는 아웃소싱 여부에 대한 결정과 자체생산 작업에 대한 일정계획결정의 두 가지 의사결정 문제를 통합한 것으로 기존의 일정계획분야의 연구들과는 차별화 된다.

최근에 본 연구에서와 같이 아웃소싱옵션을 가지는 생산시스템에서의 일정계획과 관련한 연구가 Lee and Sung[9, 10]에 의해서 수행되었다. 그들은 본 연구에서와 동일하게 총 작업완료시간과 총 아웃소싱비용을 합한 총 비용을 최소화할 수 있는 일정계획연구를 수행하였으며, 특히 Lee and Sung[10]은 일정계획비용으로 총 작업완료시간이 아닌 납기관련함수를 사용하였다. 이들의 연구는 공통적으로 자체생산설비가 단일기계(single machine)인 상황만을 고려한 것으로 본 연구에서는 이를 여러 대의 기계를 포함하는 병렬기계숍으로 확장하였다. 또한, 본 연구에서는 해당 문제가 NP-complete 영역에 속하는 난제임을 감안하여 작은 사이즈의 문제를 대상으로 한 동적계획법(dynamic programming), 분지한계법(branch-and-bound)과 같은 최적화 기법과 큰 사이즈의 문제를 위한 발견적 해법(heuristic method)을 동시에 제안하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 연구에서 다루고 있는 아웃소싱을 고려한 생산일정계획문제를 정의하고, 문제에 대한 몇 가지 우월성질을 규명한다. 제 3장에서는 제시된 우월성질을 바탕으로 최적해를 위한 동적계획법과 효율적인 휴리스틱을 제안하고, 제 4장에서는 분지한계법을 제시하였다. 제 5장에서는 제안된 휴리스틱과 분지한계법의 성능평가를 위해 다양한 수치실험의 결과를 정리하였다. 마지막으로 제 6장에서는 결론과 향후 연구방향에 대한 논의가 이루어진다.

2. 문제정의와 분석

본 연구의 대상이 되는 아웃소싱을 고려한 생산 일정 계획 문제를 정의하면 다음과 같다. 처리해야할 총 n 개의 작업(job, 또는 주문)이 존재하고, 각 작업은 m 대의 기계로 구성된 병렬설비를 통해 자체 처리되거나 외부 생산업체에 의주되어 조달된다. 작업 $i(i=1, 2, \dots, n)$ 가 자체처리될 경우에는 p_i 의 작업시간이 소요되며 해당 작업의 완료시간은 C_i 로 표시된다. 만약, 작업 i 가 외부에 아웃소싱 될 경우에는 o_i 의 아웃소싱비용이 발생하며, 이 경우 그 작업의 완료시간은 아웃소싱 리드타임인 l_i 가 된다. 여기서, C_i 는 병렬설비에 대한 생산일정 계획에 따라 결정되는 변수이지만, l_i 는 아웃소싱계약에 의해 미리 정해진 납기로서 고정된 상수로 가정된다. 이때, π 를 전체 작업 중에 아웃소싱으로 선택된 작업들의 부분집합이라고 할 때, 총 아웃소싱비용 $OC(\pi)$ 는 $\sum_{i \in \pi} o_i$ 가 된다. 또한, 본 연구에서는 추가적으로 “아웃소싱 예산제약”(outsourcing budget constraint)을 고려하고 있다. 즉, 총 아웃소싱비용 $OC(\pi)$ 는 총 가용예산 K 범위 이내로 제한되어, $OC(\pi) = \sum_{i \in \pi} o_i \leq K$ 의 제약조건이 존재한다.

본 연구의 목적함수에 해당하는 총 비용은 총 작업완료시간과 총 아웃소싱 비용의 합으로 정의된다. 여기서, 작업완료시간은 해당 주문에 대한 고객만족도와 반응성을 고려함과 동시에 자체 생산설비에서의 재공재고(WIP) 비용을 반영한 것이며 아웃소싱 비용은 작업을 위탁하는 과정에서 발생하는 추가비용을 고려한 것이다. 자체 처리되는 작업이 많으면 자체 설비의 용량제한으로 인해 총 작업완료시간이 증대할 것이고, 반대로 아웃소싱 되는 작업이 많으면 아웃소싱 비용이 증가하게 된다. 따라서 작업별로 아웃소싱 여부를 판단할 때 기업의 입장에서는 이러한 두 가지 비용간의 상보관계(trade-off)를 고려하여 전체 비용이 최소화되는 방향으로 의사결정이 이루어져야 한다. 따라서, 본 연구가 고려하는 총 비용 TC 는 다음과 같이 표현될 수 있다.

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

여기서, $\tilde{\pi}$ 는 자체 생산되는 작업들의 부분집합이고, δ 는 0과 1사이의 실수 값을 가지는 상수로서, 작업완료시간과 아웃소싱 비용간의 조화를 위해 도입한 계수이며 경영진에서 판단하는 두 비용 간의 가중치차이를 반영할 수 있도록 도입하였다.

결과적으로, 본 연구의 목적은 아웃소싱 예산제약 $OC(\pi) \leq K$ 을 만족하면서, 식 (1)의 총 비용을 최소화할 수 있도록 아웃소싱 작업집합 π 를 결정하고 자체생산 작업집합 $\tilde{\pi}$ 에 대한 병렬기계 일정계획을 수립하는 것이다. 지금까지 정의한 아웃소싱 옵션하에서 생산일정계획 문제를 서술의 편의상 앞으로 PMSO(Parallel machine scheduling with outsourcing options)라고 지칭하기로 한다.

PMSO 문제의 수리모형은 다음과 같이 표현될 수 있다.

$$\begin{aligned} & \text{Min} \\ & \sum_i (\delta o_i + (1-\delta)l_i)y_i + (1-\delta) \sum_j \sum_k \sum_i p_i (q_k - j + 1)x_{ijk} \\ & \text{s.t.} \quad \sum_j \sum_k x_{ijk} + y_i = 1, \text{ for } \forall i. \\ & \quad \sum_j \sum_i x_{ijk} = q_k, \text{ for } \forall k. \\ & \quad \sum_i o_i y_i \leq K. \\ & \quad \sum_i x_{ijk} \geq \sum_i x_{ij+1k}, \text{ for } \forall j, k. \\ & \quad x_{ijk}, y_i \in \{0, 1\} \text{ and } q_k \in Z^+ \cup \{0\}. \end{aligned}$$

수리모형에서 사용된 변수에 대한 설명은 다음과 같다.
 y_i : 작업 i 의 생산을 아웃소싱 한다면 값이 1, 그렇지 않으면 0,
 x_{ijk} : 작업 i 를 설비 k 의 j 번째 위치에서 직접 생산한다면 값이 1, 그렇지 않으면 0,
 q_k : 설비 k 에서 직접 생산하는 작업들의 총 수

위 모형의 목적식에서 두 번째 항목을 보면 결정변수들 x_{ijk} 와 q_k 의 곱의 형태로 표현되어 있으므로, 비선형 계획(Nonlinear Programming)에 해당된다고 말할 수 있으며, 이러한 모형에 대한 효과적인 해결책을 개발하는 것은 쉬운 일이 아닐 것이다.

더욱이 PMSO의 문제복잡도(problem complexity)가 경영과학 분야에서 말하는 ordinary NP-complete 영역의 문제임을 주목할 필요가 있다. 이는 Lee and Sung[10]에서 이미 증명되었으며 자체 생산설비가 단일기계인 경우에도 NP-complete이다. 따라서, 본 연구에서는 의사다항(pseudo-polynomial)의 시간복잡도(time complexity)를 가지는 동적계획법을 제안함과 동시에 작업 수가 큰 대형문제에도 근사해(near-optimal solution)를 빠른 시간에 찾을 수 있는 휴리스틱 알고리즘을 동시에 제안한다.

PMSO는 서술된 바와 같이 아웃소싱 할 작업군을 찾아내는 문제와 자체 처리할 작업들에 대한 일정계획을 도출하는 두 가지 의사결정이 혼합되어 있다. 여기서 만약 아웃소싱 할 작업군이 고정된 경우에는 두 번째

일정계획의 문제는 기존의 잘 알려진 병렬기계에서 총 작업완료시간최소화문제와 동일해 지며(Baker[2] 참조), 다음의 LPT(longest processing time)기반의 알고리즘에 의해 일정계획문제는 해결될 수 있다.

<LPT-based Algorithm>

입력자료 : 자체생산 작업집합 $\tilde{\pi}$, 설비수 m , 작업시간 $p_i(i \in \tilde{\pi})$

단계 0 : $\tilde{\pi}$ 에 속한 작업들을 p_i 값의 내림차순으로 정렬한다.

단계 1 : 정렬된 순서에 따라 각 작업들을 다음과 같은 방식으로 설비에 할당한다.

- 1-1 : 각 설비에 이미 할당된 작업들의 총 작업시간의 합이 가장 작은 설비에 할당하며,
- 1-2 : 이미 할당된 작업들의 가장 선두위치에 삽입한다.

단계 2 : $\tilde{\pi}$ 에 속한 모든 작업이 할당될 때 까지 Step 1을 반복한다.

PMSO 문제가 NP-complete 영역의 문제이나, 아웃소싱 예산제약이 존재하지 않는 특수한 경우에는 위의 LPT 기반 알고리즘을 활용한 다음의 동적계획법에 의해 다항식의 시간복잡도로 문제를 해결할 수 있다.

<DP 알고리즘 A>

재정렬(indexing) : LPT 순으로 작업들을 정렬한다.

가치함수(value function), $f_j(y_1, \dots, y_m)$: 작업 1부터 j 까지의 작업중에서 설비 $k(k=1, 2, \dots, m)$ 에 y_k 개의 작업이 할당되고 나머지 $(j - \sum_{i=1}^m y_i)$ 개의 작업들은 아

아웃소싱된 경우의 최소 목적함수 값을 나타낸다.

초기조건 : $f_0(0, \dots, 0) = 0$

최적해 : $\min_{\{(y_1, \dots, y_m) | 1 \leq y_i \leq n, i=1, \dots, m\}} f_n(y_1, \dots, y_m)$

순환식(recursive equation) :

$$f_j(y_1, \dots, y_m) = \min \begin{cases} f_{j-1}(y_1, \dots, y_{k-1}, \dots, y_m) + (1-\delta)y_k p_j \\ \text{여기서 } k = \operatorname{argmax}_{\{1 \leq j \leq m\}} y_j \quad (i) \\ f_{j-1}(y_1, \dots, y_m) + (1-\delta)l_j + \delta o_j \quad (ii) \end{cases}$$

위의 순환식에서 (i)은 LPT상에서 j 번째 작업을 직접 처리하는 경우를 의미하고, (ii)는 작업 j 를 아웃소싱하는 경우를 의미한다. 위 알고리즘의 상태(state)는 최대 n^{m+1} 이므로, 전체 알고리즘의 시간복잡도(time complexity)는 $O(n^{m+1})$ 이다.

다음의 성질 1은 일반화된 PMSO 문제에서 아웃소싱

할 작업들을 선택하기 위한 아이디어를 제공한다.

성질 1 : 다음의 조건을 만족하는 작업 j 는 아웃소싱의 고려대상에서 제외한다.

$$\delta o_j + (1-\delta)l_j > (1-\delta) \left\{ p_j \left(\left\lceil \frac{(n_\pi - q)}{m} \right\rceil + 1 \right) + \sum_{k=1}^{\lfloor j/m \rfloor - 1} p_{(j-km)} \right\} \quad (2)$$

여기서, n_π 는 아웃소싱되는 것으로 분류되지 않은 작업들의 수이며, 작업 j 는 n_π 개의 작업들 중에서 q 번째로 작은 가공시간을 갖는 작업이다. 또한, $p_{(x)}$ 는 작업시간들 중에서 x 번째로 작은 값이다.

증명 : 설비 k 에 의해 작업 j 가 생산된다고 할 때, $\psi_k(j)$ 를 트럭 k 에서 작업 j 이후에 생산되는 작업들의 집합이라 하자. 만일 작업 j 를 아웃소싱 한다면, 총비용의 증가분은 $\delta o_j + (1-\delta)l_j$ 으로 계산된다. 또한, n_π 개의 작업들을 m 대의 설비로 나누어 생산한다면, 최적 생산계획은 n_π 개의 작업들을 균등하게 나누어서 생산하게 된다. 즉, $|\psi_k(j)|$ 는 $\lceil (n_\pi - p)/m \rceil$ 의 값을 초과할 수가 없다. 여기서, $\lceil A \rceil$ 는 A 이상인 정수들 중에 최소값을 의미한다. 또한 q 번째로 작은 작업시간을 가지는 작업 j 보다 선행하는 작업들의 작업시간의 합은 $\sum_{k=1}^{\lfloor j/m \rfloor - 1} p_{(j-km)}$ 보다 클 수 없다. 결국, 총비용의 감소분의 최대값은 다음과 같다.

$$(1-\delta) \left\{ p_j \left(\left\lceil \frac{(n_\pi - q)}{m} \right\rceil + 1 \right) + \sum_{k=1}^{\lfloor j/m \rfloor - 1} p_{(j-km)} \right\}$$

따라서 식 (2)를 만족한다면 작업 j 는 아웃소싱으로 인한 비용이 이익을 초과하므로, 아웃소싱의 고려대상에서 제외되어야 한다. □

3. PMSO 문제의 해법

본 장에서는 PMSO 문제에 대한 동적계획법을 통한 최적해법과 근사해를 위한 발견적 해법을 제시하게 된다.

3.1 동적계획법

앞서, PMSO 문제가 ordinary NP-complete 영역의 문제

임을 기술한 바 있다. 이는 다항식의 시간복잡도를 가지는 알고리즘이 존재하지 않음을 의미하는 동시에 의사다항의 알고리즘이 존재함을 의미한다. 따라서, 본 절에서는 의사다항의 시간복잡도를 가지는 다음의 동적계획법을 제시하게 된다.

<DP 알고리즘 B>

재정렬(indexing) : LPT 순으로 작업들을 재정렬한다.

가치함수(Value Function) $f_j(y_1, \dots, y_m; b)$: 작업 1, ..., j 들이 최소의 비용으로 수립된 부분 생산계획. 단, 설비 k 에 작업 1, ..., j 들 중에서 y_k 개의 작업이 할당되어서, 생산되고, $(j - \sum_{i=1}^m y_i)$ 개의 작업들이 아웃소싱되며, 아웃소싱 소요비용은 b 이다.

초기조건 : $f_0(0, \dots, 0; 0) = 0$

최적해 :

$$\min_{\{y_1, \dots, y_m : b\} | 0 < b \leq K, 1 \leq y_i \leq n, i=1, \dots, m} f_n(y_1, \dots, y_m; b)$$

순환조건 :

$$f_j(y_1, \dots, y_m; b) = \begin{cases} f_{j-1}(y_1, \dots, y_{k-1}, \dots, y_m; b) + (1-\delta)y_k p_j & \text{여기서 } k = \operatorname{argmax}_{\{1 \leq j \leq m\}} y_j(i) \\ f_{j-1}(y_1, \dots, y_m; b - o_j) + (1-\delta)l_j + \delta o_j & \text{if } b - o_j \geq 0 \quad (ii) \\ \infty & \text{if } b - o_j < 0 \quad (iii) \end{cases}$$

위의 순환조건에서 (i)는 작업 j 를 직접 생산하는 경우를 의미하고, (ii)는 작업 j 를 아웃소싱하는 경우를 의미한다. (iii)은 아웃소싱 예산제약의 여유가 허락하지 않는다면, 아웃소싱은 대안으로서 고려될 수 없음을 의미한다. DP 알고리즘 B의 상태는 최대 $n^{m+1}K$ 이므로 전체 알고리즘의 시간복잡도는 $O(n^{m+1}K)$ 으로 표현될 수 있다.

3.2 발견적 해법

본 절에서는 큰 사이즈의 문제에 대해서도 효율적인 해를 제공할 수 휴리스틱을 제안하고자 한다. 제시되는 Ratio 휴리스틱은 각 작업별로 $\frac{(1-\delta)p_j}{\delta o_j + (1-\delta)l_j}$ 의 값을 계산하여, 그 값이 큰 작업일수록 아웃소싱으로 인한 이익을 얻을 것이라는 아이디어에 기반한 휴리스틱이다.

<Ratio 휴리스틱>

단계 0 : 작업들을 다음의 조건

$$\frac{(1-\delta)p_{[1]}}{\delta o_{[1]} + (1-\delta)l_{[1]}} \geq \frac{(1-\delta)p_{[2]}}{\delta o_{[2]} + (1-\delta)l_{[2]}} \geq \dots \geq \frac{(1-\delta)p_{[n]}}{\delta o_{[n]} + (1-\delta)l_{[n]}}$$

이 만족하도록 정렬한다. 또한, $q=1, B=K, n_\pi = n$ 으로 초기화한다.

단계 1 : 조건 $q > n$ 을 만족한다면, 아웃소싱되지 않는 작업들을 대상으로 LPT 기반의 알고리즘을 통해 작업순서를 정하고 알고리즘을 종료한다.

단계 2 : 만일 성질 1의 조건 (2)를 만족하거나, 조건 $o_{[q]} \geq B$ 을 만족한다면, 작업 $[q]$ 는 아웃소싱되지 않으며, q 를 $q+1$ 으로 업데이트하고, 단계 1을 다시 수행한다. 그렇지 않으면, 단계 3을 수행한다. 여기서, 작업 $[q]$ 는 n_π 개의 작업들 중에서 q 번째 작은 작업시간을 가지는 작업을 의미한다.

단계 3 : 작업 $[q]$ 는 아웃소싱 된다. 따라서, $q=q+1, B=B-o_{[q]}, n_\pi = n_\pi - 1$ 으로 업데이트하고, 단계 1로 이동한다.

4. 분지한계법

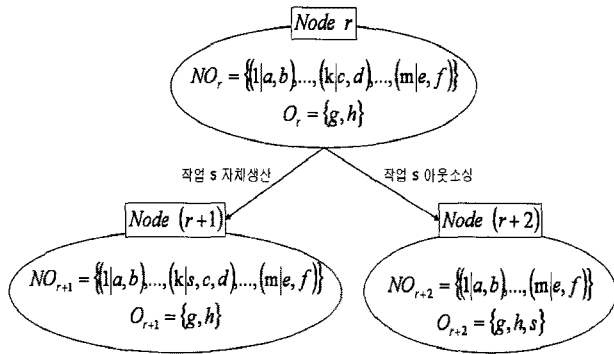
4.1 분지규칙(branching rule)

본 연구에서 제안하는 분지한계법의 각 노드는 전체 작업들의 부분집합에 대응되며, 그 중 일부작업들은 아웃소싱되고, 나머지 작업들은 직접생산으로 나뉘어지게 된다. 직접생산하는 작업들은 LPT 기반의 알고리즘에 의해 생산설비 할당과 처리순서가 결정된다. 분지단계에서는 분지할 하나의 노드(부분순열)를 선택하고 이 부분순열에 포함되지 않은 작업들 중 가장 큰 작업시간을 가지는 작업을 선택하여, 해당 작업을 아웃소싱하거나, 직접 생산하는 경우들을 고려하여 두 개의 자식노드들을 생성하게 된다.

분지한계트리에서의 각 노드들의 구조는 다음의 <그림 1>을 통해 설명한다.

그림에서 NO_r 은 노드 r 에서의 직접 생산하는 작업들의 집합을 의미하고, $(1|a, b)$ 는 설비 1에 의해서 작업 a 와 b 를 $a \rightarrow b$ 의 순서로 직접생산하고, $(k|c, d)$ 는 설비 k 에 의해서 작업 c 와 d 를 $c \rightarrow d$ 의 순서로 직접 생산함을 의미한다. 같은 방식으로 $(m|c, d)$ 는 설비 m 에 의해서 작업 e 와 f 를 $e \rightarrow f$ 의 순서로 직접 생산한다. O_r 은 노드 r 에서 아웃소싱하는 작업들의 집합을 의미한다. 따라서 $O_r = \{g, h\}$ 는 작업 g 와 h 가 아웃소싱 됨을 의미

한다. 집합 NO_r 와 O_r 에 속하지 않은 작업들 중에서, 작업시간이 가장 큰 작업 s 를 선택하여 두 개의 자식노드들 $(r+1)$ 과 $(r+2)$ 로 분지한다. 노드 $(r+1)$ 은 작업 s 를 직접 생산하는 경우의 자식노드이며, LPT 기반의 알고리즘에 의하여 설비 k 를 선정하여 할당하게 되면 $(k|s, c, d)$ 가 된다. 또한, 노드 $(r+2)$ 는 작업 s 를 아웃소싱하는 경우의 자식노드를 의미하며, 결과적으로 작업들 g, h, s 가 아웃소싱 된다.



<그림 1> 분지한계트리에서의 노드 구조

본 연구에서의 분지한계법에서는 깊이우선탐색(depth-first search)을 사용한다. 이는 탐색트리에서 가장 많은 부분작업을 가진 노드가 분지노드로 선택됨을 의미한다. 이러한 선택이 여러 노드에서 발생할 경우에는 그중에서 하한 값(lower bound)이 가장 작은 노드가 선택된다.

4.2 한계규칙(bounding rule)

한계규칙은 각 노드의 상한과 하한을 이용하여 현재 탐색중인 노드의 제거(fathoming) 여부를 결정하고, 해의 수렴을 촉진시킨다. 본 연구의 분지한계 알고리즘의 초기 상한(initial upper bound) UB 값으로는 제 3.2절에서 제안된 휴리스틱의 결과값을 활용한다.

각 노드에서의 하한값은 다음의 LB 절차로 계산된 값이 사용된다.

<LB 절차>

- 단계 0 : 생산시간을 LPT순으로 정렬한다.
- 단계 1 : 아웃소싱 비용을 오름차순으로 정렬한다.
- 단계 2 : 다음 조건을 만족하는 u 를 구한다.

$$u = \max \left\{ p \mid \sum_{j=1}^p o_{[j]} \leq K, 1 \leq p \leq n \right\}$$

단계 3 : LPT 정렬상의 첫 u 개의 작업들을 아웃소싱하며, 나머지 $n-u$ 개의 작업들을 LPT 기반의 알

고리즘을 통해 생산 일정계획을 수립한다. 그 결과로 얻어진 총 작업완료시간의 합 TDC 에 대해 $(1-\delta)TDC$ 를 계산하여 이를 하한 값으로 설정하고 종료한다.

LB 절차의 단계 2에서 계산되는 u 는 예산의 범위내에서 아웃소싱할 수 있는 최대 작업 수에 해당된다. 이때, 아웃소싱되는 u 개 작업들의 아웃소싱 비용은 고려하지 않고, $(n-u)$ 개 작업들의 생산완료시간만을 고려하여 산출하였으므로, LB 절차는 PMSO 문제의 하한값을 제공하게 된다.

π_r 을 분지한계트리의 노드 r 에서의 이미 부분적으로 결정되어 있는 생산 일정계획이라 하자. $\tilde{\pi}_r$ 를 π_r 에 속하지 않은 작업들의 집합이라고 하고, $TC(\pi_r)$ 을 부분적인 생산 일정계획 π_r 의 비용이라고 하자. 집합 $\tilde{\pi}_r$ 에 대해서 LB 절차를 적용하여 얻어진 하한값을 $LB(\tilde{\pi}_r)$ 이라 한다면, 노드 r 에서의 하한값 LB_r 은 다음의 수식을 통해서 계산되어질 수 있다.

$$LB_r = TC(\pi_r) + LB(\tilde{\pi}_r)$$

4.3 노드제거규칙(fathoming rules)

본 연구의 분지한계법에서는 두 가지 노드제거규칙을 사용한다. 첫 번째 규칙은 앞서 소개된 성질 1을 활용한다. 즉, 성질 1의 조건 (2)를 만족하는 작업이 아웃소싱되는 분지노드들은 제거됨을 의미한다. 두 번째 노드제거규칙은 다음의 성질 2에 기반한다.

성질 2 : 두 작업 i, j 이 조건 $p_i < p_j, o_i \geq o_j, l_i \geq l_j$ 들을 만족하는 상황을 고려하자. 만일, 최적 생산계획에서 작업 i 가 아웃소싱 된다면, 작업 j 도 반드시 아웃소싱 되어야 한다.

증명 : 인접작업교환(pair-wise job interchange) 방법으로 쉽게 증명이 가능하기 때문에, 구체적인 과정은 여기서 생략한다. □

5. 성능 평가

본 장에서는 제 3장에서 제안한 휴리스틱과 제 4장에서의 분지한계법의 성능을 다양한 수치실험을 통해서 분석한다. 알고리즘들은 C언어로 프로그래밍 되었고, Pentium IV 2.4GHz PC에서 실험이 수행되었다.

먼저 성능 측정에 사용되는 실험문제들의 생성규칙은

다음과 같다.

- 1) 각 작업의 직접 생산시간(p_i)은 $U(1, 10)$ 으로부터 발생시킨다. 여기서, $U(a, b)$ 는 a 와 b 를 모수로 가지는 이산형 균등분포를 가리킨다.
- 2) 작업을 아웃소싱할 때의 비용(o_i)과 아웃소싱 리드타임(l_i)은 각각 $U(1, 30)$ 과 $U(1, 100)$ 으로부터 발생시킨다.
- 3) 아웃소싱 예산 K 와 상수 δ 는 각각 $U(50, 200)$ 와 $U(0.3, 0.7)$ 으로부터 발생시킨다.

실험에 사용된 설비의 수는 2, 5, 8대의 3가지 경우이고, 작업의 수는 10, 20, 30, 40, 50, 60, 70의 7가지 경우를 고려하였다. 즉, 기계 및 작업의 수의 21가지 조합 각각에 대해서 20개의 실험문제들을 생성하여 분지한계법과 Ratio 알고리즘의 성능을 측정하였다. 실험의 결과는 <표 1>에 요약되어 있다.

<표 1> 분지한계법 및 Ratio의 성능평가 결과

m	n	Ratio 휴리스틱 Gap (%)		Branch-and-bound Algorithm			
				# of nodes		Time(s)	
		Ave.	Max	Ave.	Max	Ave.	Max
2	10	6.3	20.7	29.7	72	0.00	0.00
	20	9.4	18.5	919.5	3072	0.00	0.01
	30	12.7	23.7	126732.5	1157370	0.56	5.53
	40	13.4	26.2	27107831.2	171626326	152.88	959.99
5	10	0.0	0.0	12.7	25	0.00	0.00
	20	2.1	11.2	54.5	165	0.00	0.01
	30	3.4	9.9	657.6	5458	0.00	0.01
	40	4.7	14.7	10132.3	130691	0.04	0.56
	50	7.5	15.6	1109371.3	12684054	5.64	65.20
8	60	7.1	16.4	10275713.4	149041569	53.87	777.53
	10	0.4	3.9	13.0	24	0.00	0.00
	20	0.3	3.5	46.8	143	0.00	0.00
	30	1.0	6.5	153.7	669	0.00	0.00
	40	3.3	8.9	1631.9	9889	0.01	0.05
	50	3.4	8.8	32247.4	226558	0.16	1.13
	60	6.0	12.7	154803.6	1564791	0.81	8.19
70	5.4	12.3	1166668.9	5523768	6.83	28.77	

분지한계법에 대해서는 평균시간, 최대시간, 생성노드의 평균 수 및 최대수를 구하였다. 휴리스틱에 대해서는 최적해와의 상대편차를 구하였다. 즉, "OPT"가 분지한계법에 의해서 얻어진 최적해를 의미하고 "Heuristic"이 Ratio를 통해 얻어진 가능해의 값을 나타낸다고 할 때, 표에서 Gap(%)은

$$Gap(\%) = \frac{(Heuristic - OPT)}{OPT} \times 100$$

을 의미한다.

표에서 확인할 수 있듯이 분지한계법의 경우 비교적 큰 작업수에 대해서도 제한된 시간안에 최적해를 도출하고 있으며, 특히 설비수가 커짐에 따라 오히려 더 좋은 성능을 보임을 알 수 있다. 또한 제안된 Ratio 알고리즘의 경우도 작업수와 설비수가 커짐에 따라서 성능이 크게 떨어지지 않고 안정된 성능을 보임을 알 수 있다.

다음으로 아웃소싱 예산제한의 변화에 따른 성능변화를 살펴보았다. 설비의 수는 2대인 경우로 고정하고 가용예산 K 를 $U(50, 100)$, $U(100, 150)$, $U(150, 200)$, $U(200, 250)$ 의 4가지 경우로 나누어 실험을 진행하였으며, 그 결과가 <표 2>에 요약되어 있다. 예산제한이 커질수록 분지한계 알고리즘의 성능이 저하되었으나, 그 변화가 그리 크지 않음을 알 수가 있다.

<표 2> 아웃소싱 예산의 변화에 따른 성능평가

K	n	Ratio 휴리스틱 Gap (%)		Branch-and-bound algorithm			
				Time(s)		# of nodes	
		Ave.	Max	Ave.	Max	Ave.	Max
				Ave.	Max	Ave.	Max
{50, 100}	10	4.0	23.0	0.0	0.0	23.6	55
	25	9.4	19.4	0.1	0.3	16259.9	73288
	40	13.7	23.8	87.9	431.6	16262666.4	72988315
	Ave.	9.0	22.1	29.3	143.9	5426316.6	24353886
{100, 150}	10	5.3	30.8	0.0	0.0	23.1	52
	25	11.5	23.6	0.1	0.3	15941.5	77935
	40	16.1	24.8	122.5	757.1	22049599.5	127883523
	Ave.	10.9	26.4	40.9	252.4	7355188.0	42653836.7
{150, 200}	10	4.9	20.4	0.0	0.0	23.6	46
	25	10.9	22.4	0.1	0.6	31671.9	171185
	40	15.8	35.9	459.5	2316.4	81223727.1	490160612
	Ave.	10.5	26.2	153.2	772.3	27085140.8	163443947.7
{200, 250}	10	2.1	12.1	0.0	0.0	22.2	40
	25	10.4	18.7	0.1	0.4	20930.1	118306
	40	16.8	33.5	1144.3	5129.3	211069485.7	1018556729
	Ave.	9.8	21.4	381.5	1709.9	70363479.3	339558358.3

또한, 일정계획비용과 아웃소싱비용간의 조절계수인 δ 값의 변화에 따른 알고리즘들의 성능변화를 테스트하였다. 앞에서와 마찬가지로 설비는 2대로 고정하고, δ 값은 $U(0.3, 0.4)$, $U(0.4, 0.5)$, $U(0.5, 0.6)$, $U(0.6, 0.7)$ 의 4가지 경우를 고려하였으며, 실험결과는 <표 3>에 요약되어 있다. 대체적으로 δ 값의 변화에 따라 알고리즘 성능의 큰 변화는 없었으나, 전체적으로는 δ 값이 커질수록 알

고리즘의 성능은 약하게 향상됨을 알 수 있다.

<표 3> δ 값의 변화에 따른 성능평가

δ	n	Ratio 휴리스틱 Gap (%)		Branch-and-bound algorithm			
		Ave.	Max.	Time (s)		# of nodes	
				Ave.	Max.	Ave.	Max.
[0.3, 0.4]	10	4.4	17.5	0.0	0.0	28.0	67
	25	12.8	21.5	0.1	0.2	20913.4	57533
	40	10.7	17.9	243.5	2174.6	42783119.2	401007360
	Ave.	9.3	19.0	81.2	725.0	14268020.2	133688320
[0.4, 0.5]	10	5.6	21.9	0.0	0.0	31.1	88
	25	11.1	22.3	0.0	0.1	7175.3	26508
	40	10.9	19.3	183.5	1548.7	33520249.2	286841775
	Ave.	9.2	21.2	61.2	516.3	11175818.5	95622790
[0.5, 0.6]	10	2.3	11.1	0.0	0.0	22.5	48
	25	11.0	22.0	0.1	0.3	16139.7	90748
	40	13.1	21.7	63.4	340.8	11336600.3	58907804
	Ave.	8.78	18.3	21.1	113.7	3784254.1	19666200
[0.6, 0.7]	10	1.1	12.7	0.0	0.0	18.5	52
	25	8.9	29.9	0.0	0.2	6174.9	53442
	40	13.0	21.9	139.6	880.9	27198516.8	180571765
	Ave.	7.7	21.5	46.6	293.7	9068236.7	60208420

<표 4> 아웃소싱 비용변화에 따른 성능평가

o_j	n	Ratio 휴리스틱 Gap (%)		Branch-and-bound algorithm			
		Ave.	Max.	Time (s)		# of nodes	
				Ave.	Max.	Ave.	Max.
[10, 30]	10	1.6	8.4	0.0	0.0	17.5	39
	25	8.0	19.7	0.0	0.1	3480.8	13272
	40	9.5	18.3	13.7	87.8	2545097.9	15348052
	Ave.	6.4	15.4	4.6	29.3	849532.0	5120454.3
[20, 40]	10	0.4	5.9	0.0	0.0	13.80	34
	25	5.3	17.7	0.0	0.0	1021.6	7198
	40	5.4	13.3	0.5	4.8	110103.6	989731
	Ave.	3.7	12.3	0.3	1.6	37046.3	332321
[30, 50]	10	0.1	2.0	0.0	0.0	13.1	36
	25	3.3	10.9	0.0	0.0	178.1	62
	40	4.0	10.2	0.0	0.1	4644.9	2294
	Ave.	2.5	7.7	0.01	0.0	1612.0	7865
[40, 60]	10	0.0	0.0	0.0	0.0	10.3	16
	25	1.4	5.5	0.0	0.0	135.2	546
	40	3.2	10.0	0.0	0.0	1486.3	4666
	Ave.	1.53	5.17	0.01	0.01	543.9	1742.7

아웃소싱 비용 o_j 값에 따른 알고리즘들의 성능을 테스트하였다. o_j 는 $U(10, 30)$, $U(20, 40)$, $U(30, 50)$, $U(40, 60)$ 의 4가지 경우를 고려하였으며, 실험결과는 <표 4>에 요약되어 있다. 아웃소싱 비용이 커질수록 Ratio 알고리즘의 성능은 향상되었으며, 분지한계 알고리즘의 성능도 향상되는 것을 관찰할 수 있었다.

마지막으로, 아웃소싱 리드타임 l_j 값의 변화에 따른 알고리즘들의 성능을 테스트하였다. 아웃소싱 리드타임 l_j 는 $U(1, 100)$, $U(20, 120)$, $U(40, 140)$, $U(60, 160)$ 의 4가지 경우를 고려하였으며, 실험결과는 <표 5>에 요약되어 있다. 아웃소싱 리드타임이 커질수록 전체 알고리즘의 성능도 향상되는 것을 관찰할 수 있다.

<표 5> 아웃소싱 리드타임의 변화에 따른 성능평가

l_j	n	Ratio 휴리스틱 Gap (%)		Branch-and-bound algorithm			
		Ave.	Max.	Time (s)		# of nodes	
				Ave.	Max.	Ave.	Max.
[1, 100]	10	5.1	21.1	0.0	0.0	29.3	212
	25	9.7	30.6	0.0	0.1	6560.4	25335
	40	12.9	20.8	87.9	859.8	17044106.9	164341092
	Ave.	9.2	24.2	29.3	286.6	5683565.5	54788879.7
[20, 120]	10	0.0	0.0	0.0	0.0	14.4	20
	25	4.9	11.5	0.0	0.1	1400.9	16485
	40	12.3	18.3	15.8	74.6	3220768.3	13580925
	Ave.	5.8	9.9	5.3	24.9	1074061.2	4532476.7
[40, 140]	10	0.00	0.0	0.0	0.01	12.6	19
	25	3.0	8.8	0.0	0.01	431.9	2448
	40	8.5	18.5	2.1	27.3	447490.8	5598237
	Ave.	3.8	9.09	0.7	9.10	149311.7	1866901.3
[60, 160]	10	0.0	0.0	0.0	0.0	10.7	14
	25	0.0	0.0	0.0	0.0	70.5	246
	40	4.5	10.5	0.1	0.6	22808.2	141680
	Ave.	1.50	3.50	0.03	0.22	7629.8	47313.33

6. 결 론

본 연구에서는 예산제약이 존재하는 경우에 개별 주문의 처리가 자체생산설비를 통해 직접 수행되거나 외부설비에 위탁되어 조달되는 두 가지 선택이 모두 존재하는 경우에 처리해야 할 전체 주문에서 아웃소싱 될 주문들에 대한 선택과 자체 처리할 주문들에 대한 일정 계획을 결정하는 의사결정 문제를 고려하고 있다. 자체 설비로써 병렬습을 고려하였으며, 총 작업완료시간의 일정 계획비용과 아웃소싱비용간의 선형합을 최소화하기 위

한 동적계획법, 분지한계법, 휴리스틱 알고리즘 등을 제안하였다.

본 연구의 결과는 주문형생산방식의 시스템에서 특별히 시간경쟁이 치열하고, 아웃소싱전략이 빈번히 구사되는 생산환경에서 활용될 수 있으며, 아웃소싱 예산제약이 강하게 작용되는 환경에서 효과를 발휘할 수 있다.

기존의 일정계획연구들은 제한된 용량을 가진 자체생산설비에서 모든 작업들을 처리하는 경우만을 고려하여왔다. 본 연구는 일정계획과 아웃소싱전략을 결합한 새로운 통합 의사결정문제를 제안하고 있으며, 다양한 생산환경과 제약조건을 추가함으로써 폭넓은 추가 연구과제를 제안할 수 있다.

참고문헌

- [1] Abdel-Malek, L., Kullpattaranirun, T., and Nanthavanij, S.; "A Framework for Comparing Outsourcing Strategies in Multi-layered Supply Chains," *International Journal of Production Economics*, 97: 318-328, 2005.
- [2] Baker, K. R.; Introduction to Sequencing and Scheduling, John Wiley and Sons, Inc., New York, 118-119, 1974.
- [3] Cachon, G. P. and Harker, P. T.; "Competition and Outsourcing with Scale Economies," *Management Science*, 48 : 1314-1333, 2002.
- [4] Dekkers, R.; "Decision Models for Outsourcing and Core Competencies in Manufacturing," *International Journal of Production Research*, 38 : 4085-4096, 2000.
- [5] Huiskonen, J. and Pirttila, T.; "Lateral Coordination in a Logistics Outsourcing Relationship," *International Journal of Production Economics*, 78 : 177-185, 2002.
- [6] Kaipia, R. and Tanskanen, K.; "Vendor Managed Category Management-an Outsourcing Solution in Retailing," *Journal of Purchasing and Supply Management*, 9 : 165-175, 2003.
- [7] Kim, B.; "Dynamic Outsourcing to Contract Manufacturers with Different Capabilities of Reducing the Supply Cost," *International Journal of Production Economics*, 86 : 63-80, 2003.
- [8] Kouvelis, P. and Milner, J. M.; "Supply Chain Capacity and Outsourcing Decisions: the Dynamic Interplay of Demand and Supply Uncertainty," *IIE Transactions*, 34 : 717-728, 2002.
- [9] Lee, I. S. and Sung, C. S.; "Single Machine Scheduling with Outsourcing Allowed," *International Journal of Production Economics*, 111 : 623-634, 2008.
- [10] Lee, I. S. and Sung, C. S.; "Minimizing Due Date Related Measures for a Single Machine Scheduling with Outsourcing Allowed," *European Journal of Operational Research*, 186 : 931-952, 2008.
- [11] Ngwenyama, O. K. and Bryson, N.; "Making the Information Systems Outsourcing Decision : A Transaction Cost Approach to Analyzing Outsourcing Decision Problems," *European Journal of Operational Research*, 115 : 351-367, 1999.
- [12] Nicholson, L., Vakharia, A. J., and Erenguc, S. S.; "Outsourcing Inventory Management Decisions in Healthcare: Models and Application," *European Journal of Operational Research*, 154 : 271-290, 2004.
- [13] Offodile, O. F. and Abdel-Malek L. L.; "The Virtual Manufacturing Paradigm : The Impact of IT/IS Outsourcing on Manufacturing Strategy," *International Journal of Production Economics*, 75 : 147-159, 2002.
- [14] Tarakci, H., Tang, K., Moskowitz, H., and Plante, R.; "Incentive Maintenance Outsourcing Contracts for Channel Coordination and Improvement," *IIE Transactions*, 38 : 671-684, 2006.
- [15] Tarakci, H., Tang, K., Moskowitz, H., and Plante, R.; "Maintenance Outsourcing of a Multi-process Manufacturing System with Multiple Contractors," *IIE Transactions*, 38 : 67-78, 2006.
- [16] Wu, F., Li, H. Z., Chu, L. K., and Sculli, D.; "An Outsourcing Decision Model for Sustaining Long-term Performance," *International Journal of Production Research*, 43 : 2513-2535, 2005.
- [17] Yang, J., Qi, X., and Xia, Y.; "A Production-inventory System with Markovian Capacity and Outsourcing Option," *Operations Research*, 53 : 328-349, 2005.