

가중납기지연시간을 고려한 최적 주문처리순서에 관한 연구*

이익선** · 윤상흠***†

Scheduling Orders for Minimizing Total Weighted Tardiness*

Ik Sun Lee** · Sang Hum Yoon***

■ Abstract ■

This paper considers an order scheduling model to minimize the total weighted tardiness of orders. Each order requires different types of products. Each type of product is manufactured on its dedicated machine specified in advance. The completion time of each order is represented by the time when all the products belonging to the order are completed. The objective of this paper is to find the optimal production schedule minimizing the total weighted tardiness of a finite number of orders. In the problem analysis, we first derive a powerful solution property to determine the sequence of two consecutive orders. Moreover, two lower bounds of objective are derived and tested along with the derived property within a branch-and-bound scheme. Two efficient heuristic algorithms are also developed. The overall performances of the proposed property, branch-and-bound and heuristic algorithms are evaluated through various numerical experiments.

Keyword : Order Scheduling, Weighted Tardiness, Branch-and-Bound, Heuristic

1. 서 론

본 연구는 주문형생산방식(make-to-order)에서 고객들로부터 접수된 주문들을 처리할 때 주문간

처리순서를 결정하기 위한 주문처리 일정계획모델(order scheduling model)을 다루고 있다. 주문처리 일정계획모델은 다음과 같이 정의될 수 있다 (Leung et al.[6] 참조). 각 주문은 다양한 종류의

논문접수일 : 2007년 02월 26일 논문게재확정일 : 2008년 05월 09일

논문수정일(1차 : 2008년 03월 06일)

* 이 논문은 동아대학교 학술연구비 지원에 의하여 연구되었음.

** 동아대학교 경영학부

*** 영남대학교 경영학부

† 교신저자

제품들로 구성되고, 제품들은 병렬설비(parallel machine)를 통해 생산된다. 각 주문의 처리가 완료되는 시점은 그 주문에 속한 모든 제품들의 생산이 완료되는 시점으로 모델링 된다. 이때 제품을 생산하는 병렬설비는 지정설비모델과 유연설비모델의 두 종류로 구분된다. 먼저, 지정설비모델은 각 종류의 제품을 해당하는 지정설비(dedicated machine)에서만 생산하는 것이며, 유연설비모델은 각 설비가 모든 종류의 제품을 생산할 수 있으며, 한 종류에서 다른 종류로 생산이 전환될 때는 준비시간(setup)이 추가적으로 소요되는 경우이다. 본 연구는 지정설비모델을 고려하고 있으며, 연구의 목적은 한정된 수의 주문을 처리하는데 소요되는 총 가중납기지연시간(total weighted tardiness)을 최소화하는 주문간 처리순서를 도출하는 것이다.

주문처리 일정계획모델은 주문생산방식의 다양한 시스템에 응용이 가능하다. 먼저, Sung and Yoon [13]은 부품을 생산하는 부품생산단계(fabrication stage)와 생산된 부품을 조립하는 조립단계(assembly stage)로 구성되는 2단계 조립시스템에서 조립공정을 단순화하기 위해 부품위주설계(fabrication-driven design) 또는 모듈러설계(modular design)를 채택할 경우 부품생산단계가 조립단계에 비해 상대적으로 병목(bottleneck)공정이 되어, 일정계획의 관점에서 볼 때 전체 조립라인의 성능향상을 위해서는 부품생산단계에서의 일정계획이 중요한 과제가 되며 이는 주문처리 일정계획모델과 동일함을 기술하였다. Yang [15]은 주문처리 일정계획문제가 차량수리공정에 활용될 수 있음을 지적하였다. 여러 종류의 고장이 동시에 발생한 차량의 수리에서 각 고장의 수리는 동시에 진행될 수 있으며, 차량수리의 최종완료는 모든 고장이 수리완료되는 시점이 된다. 따라서, 이 경우 차량을 주문으로 해석하고 각 고장유형을 제품으로 대응해볼 때 고장난 여러 차량들에 대한 수리일정을 정하는 것은 주문처리순서를 찾는 것과 동일하게 된다. 또한, Peterkofsky and Daganzo [11]와 Yang and Posner

[16]는 항구에 정박한 화물선에서 병렬로 배치된 크레인을 통해 화물칸에 화물을 부리고 적재하는 과정도 주문처리 일정계획문제로 모델링 될 수 있음을 기술한 바 있다.

이러한 다양한 응용성과 확장가능성으로 인해 주문처리 일정계획모델은 가장 활발한 연구가 진행되고 있는 일정계획분야중 하나이며, 최근에 많은 연구결과들이 발표되고 있다. 먼저, Sung and Yoon [13]은 주문들의 총 가중완료시간(total weighted completion time)을 최소화하기 위한 주문처리 일정계획문제가 NP-complete임을 증명하였고, 오차한계(error bound)를 보장할 수 있는 휴리스틱 알고리즘을 제안하였다. 또한, 동일한 문제에 대해 Yoon and Sung [17]은 최적 해를 도출할 수 있는 분지한계법(branch-and-bound algorithm)을 제안하였으며, Wang and Cheng [14]과 Leung et al. [7]은 휴리스틱 알고리즘들을 추가적으로 제안하고 이들의 오차한계를 증명하였다. Ahmadi et al. [1]은 주문에 대한 가중치가 동일한 상황, 즉, 총 완료시간(total completion time)을 최소화하는 경우에도 동일하게 NP-complete임을 증명함으로써 Sung and Yoon [13]의 결과를 보다 일반화 하였다. Leung et al. [9]과 Yang and Posner [16]는 주문에 속한 각 제품들을 독립적인 지정설비가 아닌, 공통적인 병렬설비에서 생산하는 경우를 고려하고 각각 총 가중완료시간과 총 완료시간을 최소화하는 일정계획문제를 연구하였다.

이상의 문헌고찰에서 알 수 있듯이, 기존연구의 대부분은 주문처리가 완료되기까지의 주문의 완료시간이나 가중완료시간을 목적함수로 가지는 문제에 국한되어 있다. 하지만, 일정계획모델에서 완료시간의 최소화는 재공품재고(WIP) 비용을 줄이기 위한 생산자 위주의 목적함수에 해당하며, 주문처리에 따른 고객만족도를 고려하기 위해서는 각 주문에 대한 고객의 개별적인 요구납기(due date)를 고려하는 것이 바람직하다. 따라서, 본 연구에서는 각 고객주문에 대한 납기와 납기지연시간(tardiness)에 대한 가중치(weight)를 고려하여 총 가중납기

지연시간을 최소화하는 주문간 처리순서를 결정하는 일정계획문제를 고려한다. 각 주문에 대한 가중치는 단위납기지연시간에 대한 과금(penalty cost)으로 해석될 수 있다. 저자들이 파악한 바로는 이와 같이 납기를 고려한 주문처리 일정계획에 관한 기존연구로는 Leung et al.[8](이하, L's study)의 연구가 유일하다. L's study는 최대납기지연(maximum lateness)과 지연작업 개수(number of tardy jobs)의 최소화를 목적식으로 가지는 주문처리 일정계획에 대한 연구를 수행한 바 있다. 일반적으로 일정계획에서 납기지연(lateness)과 순수납기지연(tardiness)은 분석방법과 문제의 복잡도 측면에서 서로 다른 목적함수일 뿐 아니라, 본 연구에서는 최대납기지연이 아닌 총 납기지연시간을 고려하고 주문간 가중치를 추가적으로 고려함으로써, L's study에 비해 보다 일반화된 문제를 고려하고 있다.

본 연구의 대상 모델은 다음 제 2장에서 서술한 바와 같이 NP-complete 영역의 문제로써, 주문수와 설비수의 증가에 따라 문제의 복잡도가 급격히 증가하게 된다. 따라서 본 연구에서는 작은 사이즈의 문제에 대한 최적 해를 효율적으로 도출할 수 있는 분지한계 알고리즘과 큰 사이즈의 문제에 대해서는 빠른 시간 안에 효과적인 해를 제공하는 휴리스틱 알고리즘을 함께 제시한다.

본 연구의 구성은 다음과 같다. 제 2장에서는 본 연구에서 고려하고 있는 납기를 고려한 주문처리 일정계획문제에 대한 엄격한 정의가 제공된다. 또한, 분석에서 두 개의 주문 간의 우선순위관계를 규명할 수 있는 효과적인 우월성질이 제시된다. 제 3장에서는 목적함수에 대한 2가지 하한값(lower bound)을 활용한 분지한계법과 효율적인 해를 제공할 수 있는 2가지 휴리스틱 알고리즘들이 제안된다. 제 4장에서는 다양한 수치실험을 통해 제안된 우월성질, 분지한계법, 휴리스틱들의 성능을 평가한 결과를 제시한다. 마지막으로 제 5장에서는 연구의 결론과 추후연구과제에 대한 논의가 이루어진다.

2. 문제의 정의와 우월성질

서술의 편의를 위해 본 연구의 대상이 되는 총가중납기지연 최소화를 위한 주문처리 일정계획문제(order scheduling weighted tardiness problem)를 OSWT로 부르기로 한다. OSWT에서는 총 n 개의 처리해야 할 주문(또는, 작업)이 존재하고, 각 주문은 m 개의 서로 다른 제품으로 구성된다. 각 제품의 가공은 미리 정해진 전용 생산설비에서 이루어지며 이를 기존 문헌에서는 지정설비(dedicated machine)구조로 지칭하였다(Leung et al.[6, 8] 참조). 즉, 각 주문의 k 번째 제품 ($k=1, 2, \dots, m$)은 생산설비 k 에서 생산되며, 주문 j 에 속한 k 번째 제품의 가공을 위해서는 $p_{j,k}$ ($j=1, \dots, n, k=1, \dots, m$)의 가공시간이 소요된다.

각 주문의 제품 생산은 시점 0에서부터 가공이 가능하도록 준비되어 있고, 모든 생산설비는 여러 개의 주문을 동시에 가공할 수 없으며, 일단 어떤 주문에 속한 제품의 가공이 시작되면 완료될 때까지 계속된다. 각 주문의 최종 완료시간은 그 주문에 속한 m 개의 관련 제품들의 생산이 완료되는 시점으로 모델링되며 주문 j 의 완료시간을 C_j 로 표시한다. 각 주문에 대한 납기(due date)를 d_j 라 하면, 주문 j 의 납기지연시간 T_j 는 $\max\{0, C_j - d_j\}$ 로 표현되어 진다. 이때, 각 주문에는 지연시간에 대한 가중치(weight)가 존재하며 이를 w_j 로 표시할 때 전체 주문에 대한 총가중납기지연시간 TC 는 다음과 같이 표현된다.

$$TC = \sum_{j=1}^n w_j T_j = \sum_{j=1}^n w_j (\max\{0, C_j - d_j\}) \quad (1)$$

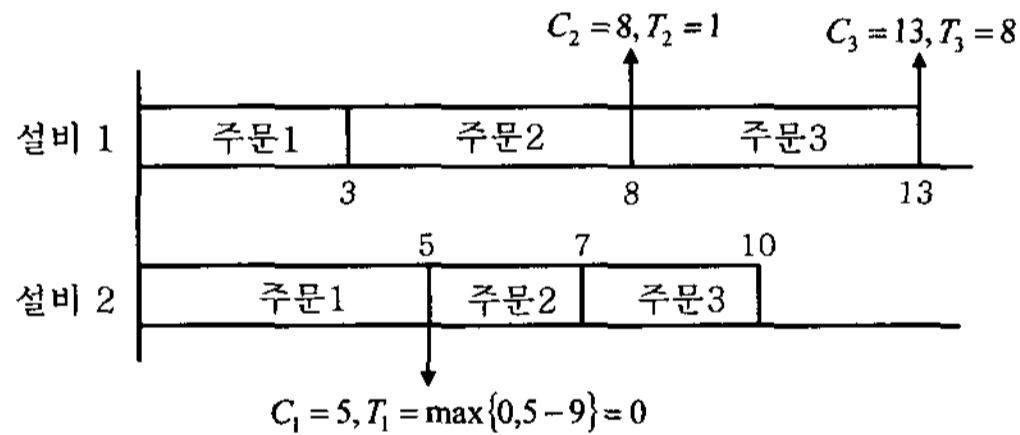
본 연구의 목적은 식 (1)의 TC 를 최소화하는 주문간 최적 처리순서를 도출하는 것이다.

기술된 OSWT문제에 대한 정확한 이해를 돕기 위해 주문이 3개이고 설비수가 2대인 간단한 예제를 제시한다. 다음의 <표 1>에 각 주문의 가공시간, 가중치, 납기에 대한 데이터가 제시되고 있다.

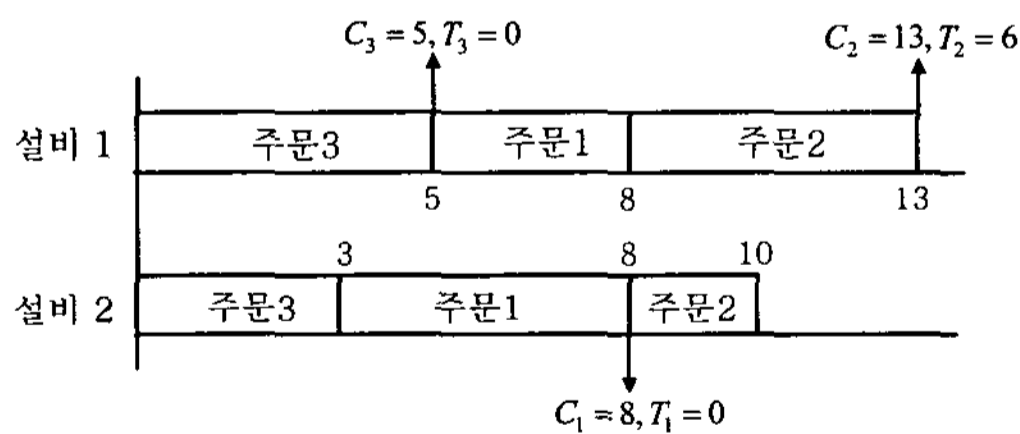
만약 두 설비에서의 주문처리를 주문 1 → 주문 2 → 주문 3의 순으로 진행할 경우 [그림 1-a]와 같이 각 주문의 납기지연은 $T_1 = 0, T_2 = 1, T_3 = 8$ 로 발생되고, 최종적인 총 가중납기지연시간은 식 (1)에 의해 $TC = 1 + 2 \times 8 = 17$ 이 된다. 만약, 주문처리 순서를 바꾸어 주문 3 → 주문 1 → 주문 2의 순으로 진행한다면, [그림 1-b]와 같이 진행되고, 납기지연은 $T_1 = 0, T_2 = 6, T_3 = 0$ 이 되어 총 가중납기지연시간은 $TC = 6$ 이 된다.

〈표 1〉 OSWT에 대한 예제 데이터

	주문 1	주문 2	주문 3
p_{i1}	3	5	5
p_{i2}	5	2	3
w_i	2	1	2
d_i	9	7	5



(a) 주문 1 → 주문 2 → 주문 3의 일정계획



(b) 주문 3 → 주문 1 → 주문 2의 일정계획

[그림 1] 두 개의 일정계획해에 대한 간트차트

문제에 대한 본격적인 해법에 들어가기 전에 먼저 OSWT문제가 NP-complete임을 주목해야 한다. 만약 모든 주문이 단일제품으로만 구성된 경우라면 OSWT문제는 잘 알려진 단일설비에서의 가중지연 일정계획문제(single machine weighted tardiness scheduling problem; SWTP)로 단순화

된다. 즉, SWTP문제는 OSWT의 특수한 경우라고 할 수 있으며, SWTP문제는 strongly NP-complete로 널리 알려져 있으므로(Pinedo[12]), 본 연구에서 다루고 있는 OSWT도 역시 strongly NP-complete임을 쉽게 유추할 수 있다. 즉, 주문수가 커짐에 따라 최적 알고리즘의 복잡도가 지수적으로 증가하는 난제의 영역에 속하므로 주문수가 큰 경우에는 주어진 시간제약 내에서 최적해(optimal solution)를 구하는 것이 어렵다는 것을 알 수 있다. 따라서 본 연구에서는 주문 수가 큰 경우에 대해서 효과적인 해를 제공할 수 있는 휴리스틱 알고리즘의 제안과 주문 수가 크지 않은 문제에 대해 최적해를 제공해 주는 효율적인 분지한계법을 동시에 제안하게 된다.

다음의 성질 1은 최적해가 순열일정계획(permutation schedule)들 중에 존재함을 밝힌 것으로, 순열일정계획은 모든 설비에서 주문간의 처리순서가 동일한 일정계획해를 의미한다. 따라서, 성질 1에 의해 OSWT문제가 $n!$ 개의 가능한 모든 순열에서 최적순열을 찾는 것과 동일함을 알게 된다.

성질 1. 순열일정계획의 집합이 우월집합(dominant set)을 구성한다.

증명. 증명은 간단한 인접작업교환(pairwise job interchange)에 의해 가능하므로, 여기서는 구체적인 증명을 생략한다. □

다음에 제시하는 성질 2는 주어진 문제에 대해 유용하게 사용할 수 있는 한 가지 우월성질(dominance property)을 제공한다. 본 우월성질은 연속해서 처리되는 두 개의 주문 간의 처리순서를 결정해 주는 것으로써, 성질자체의 의미뿐만 아니라, 3장에서 소개될 분지한계법에서 생성된 분지노드(branching node)를 제거함으로써 알고리즘의 성능을 향상시키는데 매우 유용하게 사용된다.

성질 2. 다음의 조건식 (C1)을 만족하는 두 주문 i 와 j 가 가공순서상 연속해서 처리되는 경

우에는 i 를 j 보다 앞서서 처리하는 것이 그 반대의 경우보다 우월하다.

$$w_j \Delta_i \leq w_i \Delta_j \quad (C1)$$

이때, Δ_i 와 Δ_j 는 다음의 식을 통해 계산된 값이다.

$$\Delta_i = \begin{cases} p_{i,x}, & \text{if } t_v + p_{j,v} \geq d_j \\ t_x + p_{i,x} + p_{j,x} - d_j, & \text{otherwise,} \end{cases}$$

$$\Delta_j = \begin{cases} p_{j,u}, & \text{if } t_u + p_{i,u} \geq d_i \\ t_x + p_{i,x} + p_{j,x} - d_i, & \text{otherwise,} \end{cases}$$

여기서 t_k 는 설비 k 에서 주문 i, j 보다 이전에 처리된 주문들의 생산완료시간을 나타내고, 인덱스 x, u, v 는 각각 $t_x + p_{i,x} + p_{j,x} = \max\{t_k + p_{i,k} + p_{j,k} | k = 1, \dots, m\}$, $t_u + p_{i,u} = \max\{t_k + p_{i,k} | k = 1, \dots, m\}$, $t_v + p_{j,v} = \max\{t_k + p_{j,k} | k = 1, \dots, m\}$ 가 성립할 때의 해당 설비를 나타내는 기호이다.

증명. S 를 주문 i 가 주문 j 직전에 생산되는 임의의 일정계획(sequence)이라고 하고, \tilde{S} 는 반대로 j 가 i 직전에 생산되며 i 와 j 를 제외한 나머지 주문의 처리순서는 S 와 동일한 일정계획이라고 하자. 이때 S 에서 주문 i 와 j 의 생산완료시간 C_i 와 C_j 는 인덱스 u 와 x 의 정의에 의해 각각 다음과 같이 표현된다.

$$C_i = t_u + p_{i,u}, \quad C_j = t_x + p_{i,x} + p_{j,x}.$$

따라서, S 에서 주문 i 와 j 의 납기지연 T_i 와 T_j 는 각각 다음과 같다.

$$T_i = \max\{0, t_u + p_{i,u} - d_i\},$$

$$T_j = \max\{0, t_x + p_{i,x} + p_{j,x} - d_j\}.$$

유사하게, \tilde{S} 에서의 주문 j 와 i 의 생산완료시간 \tilde{C}_j 와 \tilde{C}_i 는 인덱스 v 의 정의에 의해 다음과 같이 표현된다.

$$\tilde{C}_j = t_v + p_{j,v}, \quad \tilde{C}_i = t_x + p_{i,x} + p_{j,x}.$$

따라서, \tilde{S} 에서의 주문 j 와 i 의 납기지연 \tilde{T}_j 와 \tilde{T}_i 는 각각 다음과 같다.

$$\tilde{T}_j = \max\{0, t_v + p_{j,v} - d_j\},$$

$$\tilde{T}_i = \max\{0, t_x + p_{i,x} + p_{j,x} - d_i\}.$$

이때 성질 2의 증명은 조건 (C1)을 만족하는 경우, $w_i T_i + w_j T_j \leq w_j \tilde{T}_j + w_i \tilde{T}_i$ 의 관계가 성립함을 보이면 된다. 이를 위해 다음의 4가지 경우로 분할하여 각각 증명이 진행된다.

Case 1 : $t_u + p_{i,u} \geq d_i, t_v + p_{j,v} \geq d_j$ 인 경우.

위의 조건으로부터 $t_x + p_{i,x} + p_{j,x} \geq d_i, t_x + p_{i,x} + p_{j,x} \geq d_j$ 가 성립함을 쉽게 알 수 있다. 이 경우 조건 (C1)은 $w_j p_{i,x} \leq w_i p_{j,u}$ 를 의미하게 되며, 다음의 전개가 가능하다.

$$\begin{aligned} w_j \tilde{T}_j + w_i \tilde{T}_i &= \\ &w_j(t_v + p_{j,v} - d_j) + w_i(t_x + p_{i,x} + p_{j,x} - d_i) \\ &\geq w_j(t_v + p_{j,v} - d_j) + w_i(t_u + p_{i,u} + p_{j,u} - d_i) \\ &\quad (\text{인덱스 } x \text{의 정의에 의해서}) \\ &\geq w_j(t_v + p_{i,x} + p_{j,v} - d_j) + w_i(t_u + p_{i,u} - d_i) \\ &\quad (w_j p_{i,x} \leq w_i p_{j,u} \text{ 이므로}) \\ &\geq w_j(t_x + p_{i,x} + p_{j,x} - d_j) + w_i(t_u + p_{i,u} - d_i) \\ &\quad (\text{인덱스 } v \text{의 정의에 의해서}) \\ &= w_j T_j + w_i T_i \end{aligned}$$

Case 2 : $t_u + p_{i,u} \geq d_i, t_v + p_{j,v} < d_j$ 인 경우.

이 경우는 $t_x + p_{i,x} + p_{j,x} \geq d_i$ 와 $\tilde{T}_j = 0$ 이 성립하게 되어, 조건 (C1)에 의해 $w_j(t_x + p_{i,x} + p_{j,x} - d_j) \leq w_i p_{j,u}$ 의 관계식을 얻을 수 있으며 다음의 전개가 가능하다.

$$\begin{aligned} w_j \tilde{T}_j + w_i \tilde{T}_i &= w_i(t_x + p_{i,x} + p_{j,x} - d_i) \\ &\geq w_i(t_u + p_{i,u} + p_{j,u} - d_i) (\text{인덱스 } x \text{의 정의에 의해서}) \\ &= w_i(t_u + p_{i,u} - d_i) + w_i p_{j,u} \\ &\geq w_i(t_u + p_{i,u} - d_i) + w_j(\max\{0, t_x + p_{i,x} + p_{j,x} - d_j\}) \\ &\quad (w_i p_{j,u} \geq w_j(t_x + p_{i,x} + p_{j,x} - d_j), w_i p_{j,u} \geq 0 \text{ 이므로}) \\ &= w_j T_j + w_i T_i \end{aligned}$$

Case 3 : $t_u + p_{i,u} < d_i, t_v + p_{j,v} \geq d_j$ 인 경우.

이 경우는 $t_x + p_{i,x} + p_{j,x} \geq d_j$ 와 $T_i = 0$ 의 관계가 성립하며, 조건 (C1)에 의해 $w_j p_{i,x} \leq w_i(t_x + p_{i,x} + p_{j,x} - d_i)$ 를 의미하고 $w_j p_{i,x} \geq 0$ 이므로 다음의 전개가 성립한다.

$$\begin{aligned}
w_j \widetilde{T}_j + w_i \widetilde{T}_i &= \\
&w_j(t_v + p_{j,v} - d_j) + w_i(t_x + p_{i,x} + p_{j,x} - d_i) \\
&\geq w_j(t_v + p_{j,v} - d_j) + w_j p_{i,x} \\
&(w_j p_{i,x} \leq w_i(t_x + p_{i,x} + p_{j,x} - d_i) \text{ 이므로}) \\
&\geq w_j(t_x + p_{i,x} + p_{j,x} - d_j) \\
&(\text{인덱스 } v \text{의 정의에 의해서}) \\
&= w_j T_j + w_i T_i
\end{aligned}$$

Case 4 : $t_u + p_{i,u} < d_i$, $t_v + p_{j,v} < d_j$ 인 경우.

이 경우는 $T_i = 0$, $\widetilde{T}_j = 0$ 이며 조건 (C1)은 $w_j(t_x + p_{i,x} + p_{j,x} - d_j) \leq w_i(t_x + p_{i,x} + p_{j,x} - d_i)$ 를 의미하므로 다음의 전개가 가능하다.

$$\begin{aligned}
w_j \widetilde{T}_j + w_i \widetilde{T}_i &= w_i(\max\{0, t_x + p_{i,x} + p_{j,x} - d_i\}) \\
&(\widetilde{T}_j = 0 \text{ 이므로}) \\
&= \max\{0, w_i(t_x + p_{i,x} + p_{j,x} - d_i)\} \\
&\geq \max\{0, w_j(t_x + p_{i,x} + p_{j,x} - d_j)\} \\
&(w_j(t_x + p_{i,x} + p_{j,x} - d_j) \leq \\
&w_i(t_x + p_{i,x} + p_{j,x} - d_i) \text{ 이므로}) \\
&= w_j T_j + w_i T_i \quad (T_i = 0 \text{ 이므로})
\end{aligned}$$

결론적으로 가능한 모든 4가지 경우에 대해 모두 $w_i T_i + w_j T_j \leq w_j \widetilde{T}_j + w_i \widetilde{T}_i$ 의 관계가 성립했으므로 일정계획 S 는 \widetilde{S} 보다 우월함을 알 수 있다. \square

소개된 성질 2의 조건 (C1)은 설비수 m 과 독립임을 주목할 필요가 있다. 즉, 설비의 수가 늘어나더라도 성질을 적용할 수 있는 필요조건의 수는 늘어나지 않으며, 단일 조건 (C1)만 만족하면, 인접한 두 주문 i 와 j 의 처리순서를 최적으로 찾을 수 있으므로 매우 효과적인 우월성질이라고 할 수 있다.

3. 휴리스틱 알고리즘과 분지한계법

3.1 휴리스틱 알고리즘

본 연구에서 고려하는 OSWT문제는 strongly NP-

complete 영역의 문제이므로 보다 효율적인 방식으로 근사해(near-optimal solution)를 제공할 수 있는 휴리스틱 알고리즘의 개발은 중요하다. 또한, 본 절에서 제안하는 휴리스틱은 그 자체의 중요성뿐만 아니라 분지한계법에서 루트노드의 초기상한값(initial upper bound)을 제공함으로써 이후에 전개되는 분지한계 노드의 하한값(lower bound)과의 비교를 통해 생성된 노드의 제거(fathoming)를 가능케 하여 분지한계법의 성능을 향상시키는 역할을 한다.

본 절에서는 2개의 휴리스틱이 제안된다. 첫 번째 휴리스틱인 H1은 Kanet and Li[4]가 단일설비 가중납기지연문제(SWTP)를 위해 제안한 가중납기수정순위(weighted modified due date; WMDD)의 아이디어를 다중설비를 포함하는 주문일정계획 모델로 확장한 것이다. WMDD는 개념적으로 가중가공시간순위(weighted shortest processing time, WSPT)와 가중잔여기한순위(weighted remaining allowance; WRA)를 혼합한 것으로 다음 처리작업을 선택할 때 가중치값을 고려한 가공시간(즉, 가공시간/가중치)과 해당 작업의 가공시간 대비 납기의 여유시간(즉, 납기-가공시간)의 값이 적은 작업을 우선적으로 선택하게 된다. 단일설비환경에서는 어떤 작업을 다음 처리 작업으로 선택함으로써 발생하는 전체 완료시간에 대한 기여치(증분치)가 그 작업의 가공시간에 해당하므로 WMDD와 같은 작업선택이 유효하다. 하지만 다중기계를 포함하는 OSWT 환경에서는 그 기여치가 설비별로 다르기 때문에 설비별로 완료시간의 증분치를 고려할 필요가 있다.

따라서, 본 연구에서 제안하는 휴리스틱 H1은 WMDD를 변형한 새로운 주문할당기준치 α_j 를 활용한다. α_j 에 대한 개념은 다음과 같다. 알고리즘이 진행되면서 이전 차례에서 이미 할당되어 처리순서가 결정된 주문들의 부분일정계획(partial sequence) π 와 아직 할당되지 않은 나머지 주문들의 집합 $\bar{\pi}$ 를 가정하자. 이때 π 에 속한 모든 주문들의 설비 k 에서의 완료시점을 $T_k(k=1, \dots, m)$ 라고 할 때 T_k 는

$T_k = \sum_{j \in \pi} p_{j,k}$ 로 계산될 수 있으며, π 의 최종 완료시간 C_π 는 $C_\pi = \max_{1 \leq k \leq m} \{T_k\}$ 가 된다. 이때 아직 할당되지 않은 $\bar{\pi}$ 에 속한 주문 j 에 대해 $r_j = d_j - C_\pi$ 를 계산하면 r_j 는 주문 j 의 납기까지의 잔여시간을 의미하게 되고 이 값이 작을수록 지연이 발생할 가능성이 크므로 우선적으로 처리되는 것이 유리할 것이다. 또한, π 의 처리 직후에 주문 j 를 할당할 경우 주문 j 의 완료시간은 $C_j = \max_{1 \leq k \leq m} \{T_k + p_{j,k}\}$ 로 계산될 수 있으며, 주문 j 에 대해 계산되는 $\hat{p}_j = C_j - C_\pi$ 의 값은 주문 j 를 할당함으로써 유발되는 최종 완료시간의 증분치에 해당하므로 만약 \hat{p}_j 의 값이 납기까지의 잔여시간 r_j 를 초과한다면 지연이 발생하게 되고, 이 경우에는 r_j 를 기준으로 주문을 할당하기 보다는 \hat{p}_j 의 값이 작은 주문을 우선적으로 할당하는 것이 타당하다.

이러한 개념을 바탕으로 휴리스틱 H1에서는 현재 시점에서 할당할 주문을 선택할 때 납기까지의 잔여시간 r_j 와 완료시간에 대한 증분치 \hat{p}_j 를 동시에 고려하고, 각 주문의 가중치 w_j 를 혼합한 주문 할당기준치 α_j 를 아직 할당이 되지 않은 주문들을 대상으로 계산하여 이 값이 작은 주문을 우선적으로 할당한다. α_j 에 대한 구체적인 표현은 다음과 같다.

$$\begin{aligned} \alpha_j &= \frac{1}{w_j} (\max\{C_j - C_\pi, d_j - C_\pi\}) \\ &= \frac{1}{w_j} (\max\{\max_{1 \leq k \leq m} \{T_k + p_{j,k}\} - \max_{1 \leq k \leq m} \{T_k\}, d_j - \max_{1 \leq k \leq m} \{T_k\}\}) \end{aligned} \quad (2)$$

<휴리스틱 H1>

- 단계 0 : $\bar{\pi} = \{1, 2, \dots, n\}$, $T_k = 0 (k=1, \dots, m)$.
- 단계 1 : $\bar{\pi}$ 에 속하는 주문들에 대해 식 (2)의 α_j 를 계산하고 이 값이 가장 작은 주문 x 를 선택한다.
- 단계 2 : 주문 x 를 집합 $\bar{\pi}$ 에서 제거하고, 시점 $T_k (k=1, \dots, m)$ 를 $(T_k + p_{x,k})$ 의 값으로 업데이트

이트 한다.

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

H1에서는 납기(d_j)가 커서 납기지연이 발생할 가능성이 적은 주문임에도 불구하고 그 가중치(w_j)가 매우 클 경우에는 결과적으로 주문할당기준치 α_j 가 작아지게 되고 해당작업이 우선적으로 선택되는 단점이 발생하게 된다. 이러한 단점을 극복하기 위해 두 번째 휴리스틱 H2는 현재 시점에서 지연이 발생할 가능성이 가장 큰 주문들로만 집합 G 를 구성하고 G 에 속한 미 할당 주문들 중에서 가장 작은 $\hat{p}_j = C_j - C_\pi$ 값을 가지는 주문을 우선적으로 선택한다. 이때, G 는 어떤 기준시점 A 를 설정하고 A 이하의 납기를 가진 주문들로 구성한다. 휴리스틱 H2에서는 기준시점 A 를 정할 때 식 (2)의 α_j 를 활용하여 A 를 $\max\{T_k + p_{j^*,k}, d_{j^*}\}$ 의 값으로 설정한다. 여기서, j^* 는 $\bar{\pi}$ 에 속하는 주문 중에 가장 작은 α_j 값을 가지는 주문을 의미한다. 휴리스틱 H2를 단계별로 표현하면 다음과 같다.

<휴리스틱 H2>

- 단계 1. $A = \max\{T_k + p_{j^*,k}, d_{j^*}\}$, 여기서 j^* 는 $\bar{\pi}$ 에 속하는 주문 중에 가장 작은 α_j 값을 가지는 주문을 의미한다.
- 단계 2. 작업집합 $G = \{j | d_j \leq A\}$ 를 구성한다.
- 단계 3. G 에서 속하는 주문들 중에서 가장 작은 $\frac{1}{w_j} (\max_{1 \leq k \leq m} \{T_k + p_{j,k}\} - \max_{1 \leq k \leq m} \{T_k\})$ 를 가지는 주문 x 를 선택한다.
- 단계 4, 단계 5 : 휴리스틱 H1의 단계 2, 단계 3과 동일하다.

본 절에서 제안한 휴리스틱 H1과 H2는 전체 반복(iteration)횟수가 n 회이고 각 반복마다 최대 mn 회의 비교를 수행하게 되므로 전체적인 알고리즘의 시간복잡도는 $O(mn^2)$ 이다.

3.2 분지한계법

분지한계해법에서 한계규칙(bounding rule)은 각 노드의 상한과 하한을 이용하여 현재 탐색중인 노드의 제거 여부를 결정하고, 해의 수렴을 촉진시킨다. 본 연구에서는 초기 상한값으로 3.1절에서 제시한 두 개의 휴리스틱 알고리즘들의 목적함수 값을 구하고 그 중에서 해가 더 우수한 것을 사용한다. 즉, UB_1 와 UB_2 를 각각 휴리스틱 H1과 H2를 활용하여 얻은 목적함수값이라고 할 때, 초기노드의 상한값 UB 는 다음과 같다.

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

또한, 이후의 탐색에서 더 좋은 가능해(feasible solution)가 얻어지면 상한 값은 계속 갱신된다.

각 노드에서의 하한 값으로는 두 개의 하한 값 LB_1 과 LB_2 를 사용한다. 이의 도출과 설명을 위해 3.1절에서와 마찬가지로 다음과 같은 기호를 사용한다.

π : 현재 노드에서 이미 처리순서가 결정된 부분 순열,

n_π : π 에 포함된 주문의 수,

T_k : π 에 속한 마지막 주문의 생산설비 k 에서의 완료시점, $k = 1, \dots, m$,

$\tilde{\pi}$: π 에 속하지 않는 나머지 주문들로 구성된 임의의 부분 순열,

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

분지한계 트리의 각 노드에는 부분순열 π 가 일대일로 대응되며, 아직 처리순서가 정해지지 않은 주문들에 대한 임의의 부분순열 $\tilde{\pi}$ 를 π 의 뒤에 붙임으로써 얻어지는 임의의 전체순열 S 에 대한 총가중납기지연시간은 다음과 같다.

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

이때 $[i]$ 를 전체순열 S 에서 i 번째 순서에 위치하

는 주문을 나타낸다고 할 때, $TC(\tilde{\pi})$ 는 다음과 같이 표현되고,

$$\begin{aligned} TC(\tilde{\pi}) &= \sum_{j=n_\pi+1}^n w_{[j]} \max\{0, C_{[j]} - d_{[j]}\} \\ &= \sum_{j=n_\pi+1}^n w_{[j]} \max\{0, \max_{1 \leq k \leq m} \left\{ T_k + \sum_{i=n_\pi+1}^j p_{[i]k} \right\} - d_{[j]}\} \end{aligned} \quad (3)$$

위의 식 (3)은 다음과 같은 관계가 성립한다.

$$\begin{aligned} \text{식 (3)} &\geq \max\left\{0, \sum_{j=n_\pi+1}^n w_{[j]} \left(\max_{1 \leq k \leq m} \left\{ T_k + \sum_{i=n_\pi+1}^j p_{[i]k} \right\} - d_{[j]} \right)\right\} \\ &= \max\left\{0, \sum_{j=n_\pi+1}^n w_{[j]} \left(\max_{1 \leq k \leq m} \left\{ T_k + \sum_{i=n_\pi+1}^j p_{[i]k} \right\} \right) - \sum_{j=n_\pi+1}^n w_{[j]} d_{[j]} \right\} \\ &\geq \max\left\{0, \max_{1 \leq k \leq m} \left\{ \sum_{j=n_\pi+1}^n w_{[j]} (T_k + \sum_{i=n_\pi+1}^j p_{[i]k}) \right\} - \sum_{j=n_\pi+1}^n w_{[j]} d_{[j]} \right\} \\ &= \max\left\{0, \max_{1 \leq k \leq m} \left\{ T_k \sum_{j=n_\pi+1}^n w_{[j]} + \sum_{j=n_\pi+1}^n w_{[j]} \sum_{i=n_\pi+1}^j p_{[i]k} \right\} - \sum_{j=n_\pi+1}^n w_{[j]} d_{[j]} \right\} \end{aligned} \quad (4)$$

식 (4)의 마지막 항과 중괄호안의 첫 번째 항은 π 가 주어진 경우에 상수이다. 중괄호안의 두 번째 항은 $\tilde{\pi}$ 에 속하는 주문들을 설비 k 에서 작업시간 $p_{j,k}$ 에 대해 $WSPT(=p_{j,k}/w_j)$ 순으로 정렬할 때 최소화 될 수 있다. 따라서 $WSPT_k(\tilde{\pi})$ 가 $n-n_\pi$ 개의 주문집합 $\{p_{i,k}, w_i | i \in \tilde{\pi}\}$ 에 대해 $WSPT$ 순서의 결과로 발생하는 총가중완료시간(total weighted completion time)을 나타낸다고 할 때 식 (4)는 다음의 조건을 만족한다.

$$TC(\tilde{\pi}) \geq \max \left\{ 0, \max_{1 \leq k \leq m} \left\{ T_k \sum_{j=n_\pi+1}^n w_{[j]} + \right. \right. \\ \left. \left. WSPT_k(\tilde{\pi}) \right\} - \sum_{j=n_\pi+1}^n w_{[j]} d_{[j]} \right\}$$

따라서, 첫 번째 하한값 LB_1 은 다음과 같이 정의되고 각 노드에서 LB_1 의 계산을 위해서는 $O(mn \log n)$ 의 시간복잡도가 소요된다.

$$LB_1 = TC(\pi) + \max \left\{ 0, \max_{1 \leq k \leq m} \left\{ T_k \sum_{j \in \tilde{\pi}} w_j + \right. \right. \\ \left. \left. WSPT_k(\tilde{\pi}) \right\} - \sum_{j \in \tilde{\pi}} w_j d_j \right\} \quad (5)$$

두 번째 하한을 구하기 위해서 식 (4)를 다음과 같이 다시 표현한다.

$$\text{식 (4)} \geq \max \left\{ 0, \frac{1}{m} \sum_{k=1}^m \left\{ T_k \sum_{j=n_\pi+1}^n w_{[j]} + \right. \right. \\ \left. \left. \sum_{j=n_\pi+1}^n w_{[j]} \sum_{i=n_\pi+1}^j p_{[i]k} \right\} - \sum_{j=n_\pi+1}^n w_{[j]} d_{[j]} \right\} \quad (6) \\ = \max \left\{ 0, \frac{1}{m} \sum_{k=1}^m T_k \sum_{j=n_\pi+1}^n w_{[j]} + \frac{1}{m} \sum_{j=n_\pi+1}^n w_{[j]} \right. \\ \left. \sum_{i=n_\pi+1}^j \sum_{k=1}^m p_{[i]k} - \sum_{j=n_\pi+1}^n w_{[j]} d_{[j]} \right\}$$

식 (6)의 첫 번째 항과 세 번째 항은 상수이고, 두 번째 항은 $\tilde{\pi}$ 에 속하는 주문들을 각각의 작업시간의 합 $\sum_{k=1}^m p_{j,k}$ 에 대해 $WSPT(= \sum_{k=1}^m p_{j,k} / w_j)$ 순으로 정렬했을 때 최소화 될 수 있다. 즉, $WSPT_\Sigma(\tilde{\pi})$ 가 $n - n_\pi$ 개의 주문집합 $\left\{ \sum_{k=1}^m p_{i,k}, w_i \mid i \in \tilde{\pi} \right\}$ 에 대한 $WSPT$ 순서의 결과로 발생되는 총 가중완료시간을 나타낸다고 할 때 두 번째 하한값 LB_2 는 다음과 같이 정의된다.

$$LB_2 = TC(\tilde{\pi}) + \max \left\{ 0, \frac{1}{m} \sum_{k=1}^m T_k \sum_{j \in \tilde{\pi}} w_j + \right. \\ \left. \frac{1}{m} WSPT_\Sigma(\tilde{\pi}) - \sum_{j \in \tilde{\pi}} w_j d_j \right\} \quad (7)$$

또한, 알고리즘의 시작단계에서 각 주문 i 에 대해 $\sum_{k=1}^m p_{i,k}$ 을 미리 계산하여 저장할 경우에는 각 노드에서 LB_2 의 계산을 위해서는 $O(n \log n)$ 의 시간복잡도가 소요된다.

본 연구의 분지한계법에서는 분지한계 트리의 각 노드에서 도출된 두 가지 하한 LB_1, LB_2 를 계산하여 보다 우수한 것을 각 노드별 최종 하한으로 사용한다. 즉 각 노드에서의 최종 하한값 LB 는 다음과 같다.

$$LB = \max \{ LB_1, LB_2 \} \quad (8)$$

분지한계 트리가 전개되면서 다음에 분지할 노드의 선택은 깊이우선탐색(depth-first search)을 사용한다. 이는 탐색트리에서 가장 큰 부분순열을 가진 노드가 분지노드로 선택됨을 의미한다. 이러한 선택이 여러 노드에서 발생할 경우에는 그중에서 하한값이 가장 작은 노드가 선택된다.

지금까지 설명한 분지규칙과 한계규칙을 활용하여 전체적인 분지한계법을 단계별로 표현하면 다음과 같다. 설명에서 S^* 는 현재까지의 기대해(incumbent solution)를 나타내고 Z^* 는 그 목적함수값을 의미한다.

<분지한계 알고리즘>

단계 0. (초기화).

- 0.1. 루트노드를 생성한다.
- 0.2. 3.1절에서의 휴리스틱을 활용한 초기해를 S^* 에 대입하고 그 목적함수값은 Z^* 에 저장한다.

단계 1. (분지 및 우월성질적용).

- 1.1. 분지한계트리의 모든 노드가 탐색되었으면 알고리즘을 종료한다. 현재의 S^* 가 최적해이다.
- 1.2. 그렇지 않으면 깊이우선탐색규칙을 활용하여 현재 트리의 노드중에서 다음에 분지할 노드를 선택하고 그 자식노드들을 생성한다.

- 1.3. 새로 생성된 노드에 대해 제 2장의 성질 2가 적용되는 노드는 제거한다.

단계 2. (하한계산).

- 2.1. 단계 1에서 제거되지 않은 노드에 대해 식 (5), 식 (7), 식 (8)을 활용하여 하한값을 계산한다.
- 2.2. 계산된 하한값이 현재의 Z^* 보다 크거나 같으면 그 노드는 제거하고 그렇지 않으면 트리에 삽입한다.
- 2.3. 새로 삽입된 노드가 전체일정계획(full sequence)에 해당하면 단계 3으로 이동한다. 그렇지 않으면 단계 1로 이동한다.

단계 3. (상한값 갱신).

- 3.1. 새로 얻어진 전체일정계획의 목적함수값이 현재의 Z^* 보다 크거나 같으면 해당노드는 제거한다.
- 3.2. 그렇지 않으면 S^* 과 Z^* 를 갱신하고, 현재 트리에 속한 모든 노드들의 하한값과 갱신된 Z^* 를 비교하여 하한값이 크거나 같으면 해당 노드는 트리에서 제거한다.
- 3.3. 단계 1로 이동한다.

4. 알고리즘 성능평가

본 장에서는 제 3장에서 제안된 휴리스틱과 분지한계법의 성능을 다양한 수치실험을 통해 분석한다. 각 알고리즘들은 C언어로 프로그래밍 되었으며 실험은 Pentium IV 2.4 GHz PC에서 수행되었다. 실험을 위한 데이터는 주문의 수(n)를 9종류(8, 10, 12, 14, 16, 18, 20, 22, 24), 생산설비의 수(m)를 3종류(2, 6, 10)에 대해 생성하였다. 납기데이터는 문제의 난이도 조절을 위해 납기 범위계수(RDD; Relative range of due dates)와 납기 tightness (TF : Tardiness Factor)를 고려하여 생성하였다. RDD와 TF를 활용한 납기데이터의 생성은 납기지연이나 가중납기지연을 목적함수로 가지는 다양한 일정계획문제에서 알고리즘의 성능실험을 위해 활용되는 방법이다(Huegler and Vasko[3], Kanet

and Li[4], Maheswaran and Ponnambalam[10]). 본 실험에서는 RDD를 3종류(0.2, 0.5, 0.8), TF를 3종류(0.2, 0.5, 0.8)로 설정하였다. 주문수, 설비수, RDD, TF의 총 243가지($9 \times 3 \times 3 \times 3$)의 문제조합에 대해 각각 30문제씩, 총 7,290개의 문제를 임의로 생성하였다. 생성에 사용된 구체적인 모수는 다음과 같다.

- 1) 가공시간 $p_{j,k}$ ($j=1, \dots, n, k=1, \dots, m$)은 $U(1, 10)$ 으로부터 발생시킨다. 여기서, $U(a, b)$ 는 a 와 b 를 모수로 가지는 이산형 균등분포를 의미한다.
- 2) 납기 가중치 w_j ($j=1, \dots, n$)는 $U(1, 10)$ 으로부터 발생시킨다.
- 3) 납기 d_j ($j=1, \dots, n$)는 $U[P \times (1 - TF - RDD/2), P \times (1 - TF + RDD/2)]$ 로부터 발생시킨다. 여기서, P 는 전체 가공시간의 설비에 대한 평균값을 의미한다. 즉, $P = \left(\sum_{k=1}^m \sum_{j=1}^n p_{j,k} / m \right)$ 으로 계산된다.

납기의 경우 RDD의 값이 증가하면 각 주문의 납기에 대한 편차(분산)가 커지게 되고, 반대로 감소하면 편차가 작아지게 된다. 또한, TF의 값이 증가하면 각 주문의 납기 값이 전체적으로 작아지게 되어 지연되는 주문이 늘어나게 되고, 반대로 감소하면 납기 값이 주문의 완료시간에 비해 느슨해지는 결과가 된다.

지금까지 설명된 실험환경 하에서 먼저 분지한계법의 성능을 평가한다. 각 주문수와 설비수의 변화에 따른 각 270개씩(TF와 RDD값 3종류씩의 조합)의 문제에 대한 실험을 수행한 결과가 <표 2>에 요약되어 있다. 표에서 “탐색노드 수”는 분지한계 트리에서 생성된 전체 노드 수를 의미하고 “계산시간”은 컴퓨터의 CPU 소요시간을 나타낸다. 표에 나타난 바와 같이 600초(CPU)를 한계시간으로 설정할 때 약 20개 전후의 주문까지를 풀 수 있음을 알 수 있다. 특히, 설비의 수가 증가함에 따라 분지한계 트리의 평균 노드수가 크게 증가하지 않고 있으며, 특히

설비수가 10대로 늘어난 경우에는 20개 이상의 주문에서 한계시간을 초과한 문제가 발생하긴 했으나, 20개 이하의 주문에서는 오히려 노드수가 감소하고 있어 제안된 분지한계법의 하한값이 다중설비에 대해 효과적으로 설계되었음을 알 수 있다. 여기서, 각 설비수의 변화에 따라 노드수가 감소하고 있음에도 계산시간이 늘어나고 있는 것은 각 노드에서 우월성질의 적용여부를 체크하거나, 하한값을 계산하는 복잡

도가 늘어나기 때문으로 풀이된다.

다음으로 제 2장에서 제시된 우월성질(성질 2)이 분지한계 알고리즘의 성능에 얼마나 큰 도움을 주었는지 평가하기 위한 실험을 실시하였다. <표 3>는 성질 2를 사용하지 않았을 경우의 분지한계법의 결과를 보여주고 있다. 앞의 <표 2>의 결과와 비교해 볼 때 성질 2에 의해서 알고리즘의 성능이 크게 향상되었음을 알 수 있다.

<표 2> 분지한계 알고리즘의 성능평가

생산 설비수	주문수	계산시간			탐색 노드 수		
		Average	Median	Max.	Average	Median	Max.
2	8	<< 0.001	0.02	0.03	42	127	255
	10	<< 0.001	0.01	0.02	166	424	848
	12	0.01	0.03	0.06	602	1778	3556
	14	0.06	0.18	0.36	2614	7675	15351
	16	0.24	0.70	1.41	10265	29646	59292
	18	1.32	4.61	9.22	32733	165568	331136
	20	4.93	36.35	72.7	171040	820355	1640710
	22	23.76	116.29	232.58	711711	2846363	5692727
	24이상	(*)	(*)	(*)	(*)	(*)	(*)
6	8	<< 0.001	0.01	0.01	52	116	232
	10	0.01	0.02	0.05	161	455	910
	12	0.03	0.15	0.30	732	2327	4654
	14	0.11	0.44	0.88	2438	9501	19002
	16	0.63	3.08	6.16	10357	32419	64839
	18	2.87	49.93	99.86	37125	366068	732137
	20	12.9	95.45	190.91	152289	524332	1048665
	22이상	(*)	(*)	(*)	(*)	(*)	(*)
	10	8	<< 0.001	0.01	0.01	51	229
10		0.01	0.02	0.05	188	652	1304
12		0.03	0.11	0.22	541	1935	3871
14		0.19	2.30	4.61	2560	15871	31742
16		1.17	27.21	54.42	11422	115851	231703
18		4.69	36.86	73.72	40462	217915	435830
20이상		(*)	(*)	(*)	(*)	(*)	(*)

주) (*)표시는 600초(CPU)안에 풀 수 없는 문제가 존재하여 통계 값을 얻을 수 없는 경우임.

〈표 3〉 성질 2를 사용하지 않았을 경우의 분지한계 결과

생산 설비수	주문수	계산시간			탐색 노드 수		
		Average	Median	Max.	Average	Median	Max.
2	8	0.01	0.14	0.28	270	5202	10401
	10	0.16	1.50	2.80	6204	67242	134484
	12	4.40	72.56	145.13	133352	2572264	5144528
	14이상	(*)	(*)	(*)	(*)	(*)	(*)
6	8	0.01	0.06	0.13	233	2578	5157
	10	0.15	13.22	13.22	4136	252889	505779
	12	4.82	178.02	356.03	93230	4172937	8345875
	14이상	(*)	(*)	(*)	(*)	(*)	(*)
10	8	0.01	0.10	0.20	207	3128	6256
	10	0.18	2.73	5.47	3179	57734	115468
	12	5.05	241.99	483.98	72454	4029674	8059347
	14이상	(*)	(*)	(*)	(*)	(*)	(*)

마지막으로, RDD와 TF값의 변화에 따른 성능 변화를 살펴보기 위한 실험을 수행하였다. 이 실험은 설비수를 2대, 주문수를 20개로 고정하여 수행한 것으로, 결과는 <표 4>에 요약되어 있다. 납기 데이터의 산포가 커지거나(RDD가 증가), 총 가공 시간에 대한 상대적 인 납기의 크기가 작아지는 경우(TF가 증가)에도 분지한계가 효과적으로 대응하고 있음을 알 수 있다. 또한, RDD와 TF에 대해 공통적으로 값이 0.5인 경우에 알고리즘의 성능이

저하되는 것을 관찰할 수 있다.

다음으로 휴리스틱에 대한 성능평가를 수행하였다. 결과는 <표 5>와 <표 6>에 요약되어 있다. 표에서 Gap(%)은 최적해 대비 휴리스틱의 상대편차(relative deviation)를 나타내기 위한 척도로써 다음과 같이 계산된다.

$$Gap = \frac{TC(S_H) - TC(S^*)}{\left(\sum_{j=1}^n w_j \sum_{k=1}^m p_{j,k} \right) / m} \times 100 \quad (9)$$

〈표 4〉 RDD와 TF 값의 변화에 따른 분지한계의 성능평가

RDD	TF	계산시간			탐색 노드 수		
		Average	Median	Max.	Average	Median	Max.
0.2	0.2	1.03	8.51	17.02	57863	470151	940303
	0.5	13.31	18.44	30.33	470183	740839	1381679
	0.8	0.38	1.02	2.02	3904	10770	21487
0.5	0.2	2.79	8.77	17.55	164494	519719	1039438
	0.5	14.08	37.97	72.70	424137	852157	1640710
	0.8	0.46	1.81	3.61	5186	21688	43370
0.8	0.2	2.75	9.73	19.47	168426	617421	1234842
	0.5	8.72	12.80	24.97	234957	545441	1081933
	0.8	0.84	2.17	4.30	10214	41667	82947

여기서, $TC(S_H)$ 는 해당 휴리스틱을 통해 구해진 해의 총 가중지연시간을 나타내고, $TC(S^*)$ 는 분지한계법을 통해 구한 최적해의 총 가중지연시간을 의미한다. 일정계획문제에서 휴리스틱에 대한 성능평가를 위해 많이 사용하는 편차로는 $Gap2 = TC(S_H) / TC(S^*)$ 나 $Gap3 = [TC(S_H) - TC(S^*)] / TC(S^*)$ 를 생각할 수 있으나, Baker[2]나 Kim[5]에서 언급한 바와 같이 목적함수가 납기지연시간인 경우에는 $Gap2$ 나 $Gap3$ 를 사용할 경우 분모에 있는 $TC(S^*)$ 의 값이 0에 가까운 값을 가질 수 있으며, 이 경우 휴리스틱의 성능을 과소평가(underestimate)할 가능성이 존재하게 된다. 따라서, 본 실험에서는 식 (9)와 같이 분모에 $TC(S^*)$ 대신 주문에 대한 평균가중가공시간으로 대체한 새로운 성능평가척도를 사용하고자 한다. 휴리스틱이 최적해를 찾을 경우에는 Gap 의 값은 0을 가지게 된다. 또한, Gap 의 값이 100%가 되면 식 (9)에서 분자와 분모에 각각

$n \sum_{i=1}^n w_i$ 의 값을 나누어도 동일한 값이 되므로, 휴리스틱의 해가 최적해에 비해 (가중치를 고려한)평균 주문가공시간만큼의 차이가 발생했음을 의미하게 된다.

<표 5>에서는 생산설비수와 주문수의 조합에 대해 발생된 임의의 270문제에 대한 Gap 의 평균값과 최대값을 보여주고 있으며, 표에서 "NO"는 270개의 문제중에 최적해를 찾아낸 횟수를 나타낸 것이다. 또한, 휴리스틱 H1과 H2에 대한 개별적인 성능뿐 아니라 각 문제에 대해 두 알고리즘을 모두 적용하여 더 나은 해를 선택한 경우에 대한 결과(H1&H2)도 나타나 있다. 이 값은 실질적으로 분지한계에서 초기노드의 상한값에 대한 성능을 나타낸 것으로도 해석된다. 표에서 나타난 바와 같이 두 알고리즘이 모두 최적해를 찾아내는 횟수 측면에서 매우 우수한 성능을 보이고 있으나, Gap 이 매

<표 5> 휴리스틱의 성능평가

생산 설비수	주문수	H1			H2			H1&H2		
		Average Gap(%)	Max. Gap(%)	NO	Average Gap(%)	Max. Gap(%)	NO	Average Gap(%)	Max. Gap(%)	NO
2	10	3.76	55.93	109	3.48	30.43	118	3.11	30.43	125
	12	4.03	31.30	105	3.89	31.30	101	3.59	31.30	115
	14	4.59	40.22	78	4.40	40.22	85	4.03	40.22	93
	16	4.77	36.07	90	4.79	36.07	81	4.34	36.07	96
	18	4.33	21.12	89	4.22	20.17	86	3.74	19.22	98
	20	4.89	36.86	88	4.77	28.72	83	4.35	28.37	95
6	10	4.42	25.70	88	4.39	25.70	82	4.12	25.70	90
	12	4.61	27.27	87	4.82	27.27	81	4.40	27.27	93
	14	5.03	42.04	71	5.11	42.04	73	4.79	42.04	78
	16	5.58	47.66	68	5.83	47.66	66	5.26	47.66	75
	18	5.81	53.04	60	5.89	30.82	55	5.34	30.82	62
	20	5.95	36.99	67	5.81	30.57	67	5.27	29.39	74
10	10	4.25	34.58	84	4.41	34.58	80	4.14	34.58	85
	12	4.86	28.63	70	4.82	31.26	67	4.58	28.63	72
	14	4.54	29.28	73	4.46	30.04	79	4.23	29.28	80
	16	4.94	42.06	62	4.74	38.96	71	4.47	38.96	73
	18	6.79	41.40	58	6.88	41.40	61	6.56	41.40	65

〈표 6〉 RDD와 TF의 변화에 따른 휴리스틱의 성능평가

RDD	TF	H1			H2			H1&H2		
		Average Gap(%)	Max. Gap(%)	NO	Average Gap(%)	Max. Gap(%)	NO	Average Gap(%)	Max. Gap(%)	NO
0.2	0.2	0.11	2.52	27	0.11	2.02	25	0.08	2.02	27
	0.5	6.03	23.59	4	6.16	24.83	2	5.84	23.59	4
	0.8	7.43	36.86	0	7.26	22.86	2	6.1	22.86	2
0.5	0.2	0.19	4.11	27	0.43	4.11	23	0.19	4.11	27
	0.5	7.78	21.44	3	7.22	22.81	5	6.79	21.44	5
	0.8	7.77	30.73	2	7.39	22.57	1	6.78	22.57	2
0.8	0.2	0.66	6.94	20	0.77	6.94	21	0.63	6.94	22
	0.5	7.53	20.15	0	7.69	20.15	1	7.11	20.15	1
	0.8	6.52	28.37	5	5.92	28.72	3	5.59	28.37	5

우 큰 경우가 소수의 문제에서 발생함으로써 전체적인 평균 Gap의 값을 상승시키고 있음을 알 수 있다. 또한, 각 알고리즘이 작업수나 설비수의 증가에 따라 성능이 크게 나빠지지 않고 있음을 확인할 수 있다.

〈표 6〉은 RDD와 TF값의 변화에 따른 휴리스틱의 성능변화를 보여 주고 있다. 실험은 설비수와 주문수를 각각 2와 20으로 고정된 상태에서 진행하였으며, RDD와 TF값의 각 조합에 대해 30문제씩을 실험한 결과를 요약한 것이다. RDD와 TF에 대해 공통적으로 값이 커질 수록 알고리즘의 성능이 저하되는 것을 관찰할 수 있으며, 분지한계법의 경우와 마찬가지로 두 값이 0.5인 경우에 가장 성능이 저하된다. 이는 여타 납기지연시간을 목적함수를 가지는 문제를 연구한 기존연구에서도 많이 보고된 내용이며, 본 연구의 결과로도 이를 확인할 수 있다.

5. 결론 및 추후연구

본 연구는 고객주문이 여러 종류의 제품으로 구성되는 상황에서 병렬로 배치된 전용설비를 통해 주문을 처리할 경우의 생산일정계획문제를 다루었다. 각 주문에 대한 납기를 고려하고, 각 주문별로

지연시간에 대한 중요도를 달리 부과할 수 있도록 허용하여, 총 가중납기지연시간을 최소화하기 위한 분지한계법과 휴리스틱 알고리즘을 제안하였다.

본 연구의 확장연구로는 납기에 대한 가중치를 고려하지 않고, 목적함수를 단순 납기지연시간의 합으로 구성하는 것을 고려할 수 있다. 이 경우는 문제가 단순화되어 본 연구의 결과에 비해 보다 다양한 우월성질과 이를 활용한 효과적인 알고리즘을 개발할 수 있을 것으로 사료된다. 또한, 주문처리 이후에 주문에 대한 고객으로의 배송단계를 추가하여 전체 구조를 2단계로 구성하는 모형도 고려할 수 있다. 하지만, 이 경우는 전체구조가 부분적으로 2단계 흐름라인(flowshop)을 포함하고 있어, 흐름라인에서의 납기지연을 목적으로 하는 일정계획문제에 대한 고찰이 선행되어야 할 것으로 사료된다.

참 고 문 헌

- [1] Ahmadi, R., U. Bagchi, and T.A. Roemer, "Coordinated Scheduling of Customer Orders for Quick Response," *Naval Research Logistics*, Vol.52(2005), pp.493-512.
- [2] Baker, K.R., "Heuristic Procedures for Scheduling Job Families with Setups and the

- Due Dates," *Naval Research Logistics*, Vol. 46(1999), pp.978-991.
- [3] Huegler, P.A. and F.J. Vasko, "A Performance Comparison of Heuristics for the Total Weighted Tardiness Problem," *Computers and I.E.*, Vol.32(1997), pp.753-767.
- [4] Kanet, J.J. and X. Li, "A Weighted Modified Due Date Rule for Sequencing to Minimize Weighted Tardiness," *Journal of Scheduling*, Vol.7(2004), pp.261-276.
- [5] Kim, Y.-D., "Heuristics for Flowshop Scheduling Problems Minimizing Mean Tardiness," *Journal of the Operational Research Society*, Vol.44(1993), pp.19-28.
- [6] Leung, J.Y. T., H. Li, and M. Pinedo, "Order Scheduling Models with Applications in Practice," *Proceedings of the First Multidisciplinary International Conference on Scheduling : Theory and Applications(MISTA)*, Nottingham(2003), pp.2-10.
- [7] Leung, J.Y. T., H. Li, and M. Pinedo, "Order Scheduling in an Environment with Dedicated Resources in Parallel," *Journal of Scheduling*, Vol.8(2005), pp.355-386.
- [8] Leung, J.Y. T., H. Li, and M. Pinedo, "Scheduling Orders for Multiple Product Types with Due Date Related Objectives," *European Journal of Operational Research*, Vol. 168(2006), pp.370-389.
- [9] Leung, J.Y. T., H. Li, and M. Pinedo, "Approximation Algorithms for Minimizing Total Weighted Completion Time of Orders on Identical Machines in Parallel," *Naval Research Logistics*, Vol.53(2006), pp.243-260.
- [10] Maheswaran, R. and S.G. Ponnambalam, "An Investigation on Single Machine Total Weighted Tardiness Scheduling Problems," *International Journal of Advanced Manufacturing Technology*, Vol.22(2003), pp.243-248.
- [11] Peterkofsky, R.L. and C.F. Daganzo, "A Branch and Bound Solution for the Crane Scheduling Problem," *Transportation Research*, Vol.24B(1990), pp.159-172.
- [12] Pinedo, M., *Scheduling : Theory, Algorithms, and Systems*, Prentice Hall, Englewood Cliffs, NJ, 1995.
- [13] Sung, C.S. and S.H. Yoon, "Minimizing Total Weighted Completion Time at a Pre-assembly Stage Composed of Two Feeding Machines," *International Journal of Production Economics*, Vol.54(1998), pp.247-255.
- [14] Wang, G. and T.C.E. Cheng, "Customer Order Scheduling to Minimize Total Weighted Completion Time," to appear in *Omega* (2006).
- [15] Yang, J., *Scheduling with Batch Objectives*, Ph.D. Thesis, Industrial and Systems Engineering Graduate Program, The Ohio State University (1998).
- [16] Yang, J. and M.E. Posner, "Scheduling Parallel Machines for the Customer Order Problem," *Journal of Scheduling*, Vol.8(2005), pp.49-74.
- [17] Yoon, S.H. and C.S. Sung, "Fixed Pre-assembly Scheduling on Multiple Fabrication Machines," *International Journal of Production Economics*, Vol.96(2005), pp.109-118.