Journal of the Society of Korea Industrial and Systems Engineering
Vol. 33, No. 4, pp.1－9, December 2010.

재투입이 존재하는 2단계 흐름공정에서 총 작업 흐름시간을 최소화하는 분지한계방법

최성우* · 심상오**†

*경기대학교 경영학과
**한밭대학교 경영학과

Branch and Bound Algorithm for Two-Machine Reentrant Flowshop with the Objective of Minimizing Total Flowtime

Choi Seong-Woo* · Shim Sang-Oh**†

*Department of Business Administration, Kyonggi University
**Department of Business Administration, Hanbat National University

In this paper, we consider a two-machine re-entrant permutation flowshop scheduling problem with the objective of minimizing total flowtime, and suggest branch and bound algorithms for the scheduling problem. In this scheduling problem, each job must be processed twice on each machine, that is, each job should be processed on the two machines in the order of machine 1, machine 2 and then machine 1 and machine 2. In this research, based on the results of existing researches for re-entrant permutation flowshop scheduling problems, various dominance properties, lower bound and heuristic algorithm are suggested for the problem, and those are used to develop branch and bound algorithms. In the computational experiments for evaluation of the performance of the algorithms, the suggested branch and bound algorithms are tested on randomly generated test problems and results are reported.

Keywords : Scheduling, Re-Entrant Flowshop, Branch And Bound, Heuristics

1. 서 론

본 연구는 재투입(re-entrant)이 존재하는 2단계(two-machine) 순열흐름공정(permutation flowshop)에서 총 작업 흐름시간(flowtime)의 최소화를 목적식으로 하는 일정계획(scheduling) 문제를 고려한다. 재투입이 존재하는 2단계 순열흐름공정에서, 각 작업은 총 4개의 공정을 수행해야 한다. 즉, 각 작업은 흐름공정에서 첫 번째 설비, 두 번째 설비에서 순차적으로 공정이 진행된 후에, 다시 첫 번째, 두 번째 설비에서 재작업이 수행되어야 한다[2, 4].

재투입이 존재하는 2단계 순열흐름공정은 다음과 같이 매우 다양한 제조시스템에서 발견될 수 있다. 섬유염색공장에서 옷감들은 염색공정과 건조공정의 두 연속 공정을 수행해야 되며 일반적으로 짙은 염색을 위해 염색과 건조를 두 번씩 반복 수행하고, 거울제조공장에서는 거울에 비칠 물체의 선명한 형상을 위해 거울 뒷면의 칠(painting)공정과 건조공정을 보통 두 번씩 반복한다[2, 3,

논문접수일 : 2010년 05월 06일　　논문수정일 : 2010년 07월 22일　　게재확정일 : 2010년 07월 25일
† 교신저자 soshim@hanbat.ac.kr
※ 본 논문은 2009년도 한밭대학교 교내학술연구비의 지원을 받았음.

4]. 또한, PCB(printed circuit board) 제조공장에서는 트랜지스터(transistor)와 저항기(resistor)와 같은 전자 부품 칩들이 PCB 위에 삽입되는 공정이 부품의 종류에 따라 자동조립설비와 수동조립선반에서 각각 수행되며, PCB의 양면(윗면과 아랫면)을 대상으로 각각 진행되기 때문에 자동조립설비와 수동조립선반을 을 각각 순차적으로 두 번씩 방문해야 하는 재투입이 존재하는 2단계 순열흐름공정이 존재한다[12]. 본 일정계획 문제의 목적식을 총 작업 흐름시간의 최소화로 설정한 이유는 첫 번째 공정 앞, 그리고 공정 사이, 공정 완료 후 재공을 보관할 버퍼(buffer) 사이즈가 충분하지 않은 경우가 종종 발생하기 때문이다.

본 연구에서 고려하고 있는 재투입이 존재하는 2단계 순열흐름공정의 일정계획 문제는 Choi and Kim[2, 4]의 일정계획 문제와 목적식을 제외하고 다음과 같이 모두 동일하다. 즉, 설비(공정) 1, 설비 2, 그리고 설비 1, 설비 2의 순서로 두 대의 설비에서 공정을 진행해야 하는 n개의 job들이 존재한다고 가정하고, 임의의 job이 첫 번째로 흐름공정을 방문했을 때와 두 번째로 흐름공정을 방문했을 때 각각 first-pass sub-job과 second-pass sub-job으로 정의한다. 따라서, 본 일정계획 문제에서는 가상으로 $2n$개의 sub-job들이 존재한다고 생각할 수 있고, 각 job의 first-pass sub-job과 second-pass sub-job 사이에는 선후관계 제약조건이 존재하는 문제로 재정의 할 수 있다. 즉, 임의의 job에 대해서, 그의 first-pass sub-job이 설비 2에서 공정을 완료하기 전까지 그와 관련된 second-pass sub-job은 설비 1에서 공정을 시작할 수 없다는 제약이 존재한다. 본 연구에서는 이와 같은 제약을 first-pass sub-job과 second-pass sub-job의 선후제약조건이라고 칭하도록 하겠다. 또한, 임의의 job은 그의 second-pass sub-job이 설비 2에서 완료 되었을 때 작업이 완료된 것으로 간주한다. 즉, 본 일정계획 문제는 모든 sub-job들이 위와 같은 제약조건이 존재하는 것을 제외하고 모든 sub-job들이 설비 1, 2를 모두 순차적으로 방문하는 일반적인 2단계 흐름공정 문제와 동일하다.

본 일정계획 문제는 다음과 같은 가정을 전제로 한다 [2, 4].

1) 일정계획 수립 시초에 일정계획 수립 대상이 되는 job들의 집합이 정해져 있다.
2) 어떠한 job도 일단 공정이 시작되면 공정중간에 멈출 수 없다.
3) 설비는 고장 나지 않는다.
4) job들이 임의의 설비에서 공정을 수행하기 전 작업준비(set-up)시간이 필요하지 않다.

총 작업흐름시간의 최소화를 목적식으로 하는 흐름공정과 관련된 일정계획 문제에 대한 기존 연구는 무수히 존재한다[5, 8, 9, 13]. 재투입이 존재하는 흐름공정과 관련된 일정계획 문제에 대해서는 다음과 같은 연구결과가 존재한다. Graves et al.[7]은 반도체 공정 중 fab공정을 재투입이 존재하는 흐름공정으로 정의하고 생산성을 높일 수 있는 효율적인 휴리스틱 방법론을 개발하였고, Demirkol and Uzsoy[6]는 작업준비시간(set-up time)과 재투입이 존재하는 흐름공정을 대상으로 최대납기지연을 최소화 할 수 있는 방법론을 개발하였다. Pan and Chen[12]은 재투입이 존재하는 2단계 순열흐름공정에서 총 작업흐름시간을 최소화하는 일정계획 문제가 "*NP*-hard"임을 증명하였고 휴리스틱 방법론과 MIP(mixed integer programming) 모델을 제시하였다. 또한, Choi and Kim[2, 3, 4]는 재투입이 존재하는 2단계와 다단계 순열흐름공정의 일정계획 문제에 대하여 총 작업완료시간(makespan)과 총 납기지연(tardiness)을 최소화 할 수 있는 분지한계방법론과 다양한 휴리스틱 방법론을 개발하였다.

본 연구는 재투입이 존재하는 2단계 순열흐름공정에서 총 작업흐름시간의 최소화를 목적식으로 하는 일정계획 문제를 다룬다. 본 문제는 다음과 같이 쉽게 "*NP*-hard"임을 증명할 수 있다. Garey and Johnson(1979)은 2단계 흐름공정에서 총 작업흐름시간을 최소화하는 일정계획 문제는 "*NP*-hard"임을 이미 증명하였다. 2단계 흐름공정에서 총 작업흐름시간을 최소화하는 일정계획 문제는 본 연구에서 다루는 일정계획 문제의 특수한(special) 경우이다. 즉, 본 일정계획 문제에서 모든 second-pass sub-job들의 공정시간들이 설비 1과 설비 2에서 모두 0(zero)인 경우가 일반적인 2단계 흐름공정 문제이기 때문에, 본 연구에서 다루고 있는 일정계획 문제도 "*NP*-hard"이다.

본 일정계획문제의 최적해(optimal solution)를 구하기 위하여 Choi and Kim[4]의 2단계 순열흐름공정에서 총 납기지연의 최소화를 목적으로 하는 일정계획 문제를 대상으로 개발된 두 개의 최적해 우월성질(dominance property)들이 수정 없이 총 작업흐름시간의 최소화를 목적으로 하는 본 일정계획 문제에 적용 가능함을 증명하고, 추가로 한 개의 최적해 우월성질을 개발하고 본 연구의 분지한계방법론에 이용하였다. 또한, Choi and Kim[4]의 문제에서 하한값을 구하기 위해 개발된 두 가지 성질들을 이용하여 본 일정계획 문제의 하한값(lower bound)을 제안하고, Choi and Kim[3]의 재투입이 존재하는 다단계 순열흐름공정에서 총 작업완료시간의 최소화를 목적으로 하는 일정계획 문제를 대상으로 개발된 휴리스틱 방법론을 이용하여 초기 상한값을 구한 뒤, 이들을 본 연구에서 제안한 분지한계방법론에 이용하였다.

2. 우월성질(Dominance Properties)

본 장에서는 본 연구에서 제안한 분지한계(B&B) 방법론에서 사용될 최적해 우월성질을 보여준다. 본 연구에서 제안된 최적해 우월성질을 이용하여 임의의 부분(partial) 스케쥴(schedule)이 다른 부분 스케쥴에 지배(dominate)될 수 있다는 것을 확인할 수 있고 제안된 분지한계 방법론의 진행절차 중 더 이상 고려할 필요가 없는 가지(branch), 즉, 부분 스케쥴에 해당하는 노드(node)들을 제거할 수 있다[2, 4] 즉, 만약 임의의 부분 스케쥴(a)로부터 완성된 최적의 스케쥴(ax)이 또 다른 부분 스케쥴(b)로부터 완성된 최적의 스케쥴(bx)보다 좋지 않은 스케쥴이라면 그 부분 스케쥴(a)은 다른 부분 스케쥴(b)에 의해서 지배되며 더 이상 고려할 필요가 없다는 뜻이다.

본 장에서는 먼저 재투입이 존재하는 2단계 순열(permutation)흐름공정 문제를 Choi and Kim[2, 4]의 연구결과와 동일하게 다음과 같이 정의한다. 즉, 본 2단계 순열(permutation)흐름공정의 일정계획 문제에서 각 subjob들(선후관계제약이 존재하는 first-pass 또는 second-pass)의 공정순서(sequence)는 설비 1과 설비 2에서 동일하다. 본 총 작업흐름시간의 최소화를 목적으로 하는 2단계 순열흐름공정의 일정계획 문제에서 각 sub-job들의 공정순서(sequence)가 정해질 경우 각 job들의 스케쥴 및 작업흐름시간 또한 쉽게 얻어질 수 있기 때문에 "공정순서(sequence)"와 "스케쥴(schedule)"은 본 연구에서 동일한 의미로 사용할 것을 미리 밝혀둔다. 일반적으로 많은 제조라인에서는 연속된 두 대의 설비로 구성된 흐름공정에서 두 대의 설비사이에 제한된 저장 공간과 기술적인 문제로 인해 job들의 다른 공정순서로 각 설비에서 공정을 진행시키는 것이 불가능한 경우가 많기 때문에 본 연구에서도 위와 같이 순열흐름공정 혹은 순열공정순서만을 고려한다. 다음은 본 연구에서 사용할 기호와 목적식을 의미한다.

- i^k : job i의 kth-pass sub-job($i = 1, 2, \cdots, n$, and $k = 1, 2$)
- $I^\bullet$: job i의 sub-job($i^\bullet$는 first-pass sub-job 혹은 second-pass sub-job이 될 수 있음)
- $p_{i^k j}$: sub-job i^k의 설비 j에서의 공정시간($j = 1, 2$)
- σ : sub-job들의 부분스케쥴(partial sequence) 혹은 부분공정순서(partial sequence)
- $c_{i^k j}(S)$: 스케쥴 S에서 설비 j에서 sub-job i^k의 공정완료시간
- $C_j(S)$: 스케쥴 S에 포함되어 있는 sub-job들의 설비 j에서 총 작업완료 시간
- π : σ에 포함되지 않은 sub-job들의 임의의 부분스케쥴
- $\sigma i_1^\bullet i_2^\bullet, \cdots, i_l^\bullet$: σ뒤에 sub-job $i_1^\bullet, i_2^\bullet, \cdots i_l^\bullet$가 순서데로 스케쥴된 부분스케쥴
- $\sigma\pi$: σ뒤에 π가 바로 수립된 부분스케쥴
- $F(S)$: 스케쥴 S의 총 작업흐름시간, 즉, 본 스케쥴링 문제에서 스케쥴 S의 목적식, $\sum_{(i^2)^\forall \in S} c_{i^2 2}(S)$

스케쥴 $S(i^2 \in S)$에서 job i의 흐름시간(작업완료시간)은 설비 2에서 job i의 second-pass sub-job i^2의 작업완료시간으로 정의될 수 있으며, $c_{i^2 2}(S)$의 식으로 표현 될 수 있다.

앞서 설명한바와 같이 어떠한 스케쥴(공정순서)이 만약 다른 스케쥴에 의해서 지배(dominate)된다면 해당 스케쥴(공정순서)는 다음과 같은 최적해 우월성질(dominance property)에 의해서 제거될 수 있다. 본 연구에서는 총 3개의 최적해 우월성질을 제안하였다.

Proposition 1은 서로 다른 두 개의 job들의 근접한(adjacent) first-pass sub-job들에 대한, Proposition 2는 서로 다른 두 개의 job들의 근접한 first-pass sub-job과 secondpass sub-job들에 대한, Proposition 3은 서로 다른 두 개의 job들의 근접한 second-pass sub-job들에 대한 각각의 우월성질이다. 우선, Proposition 1과 Proposition 2는 Choi and Kim[4]의 2단계 순열흐름공정에서 총 납기지연의 최소화를 목적으로 하는 일정계획문제를 위해 개발 된 최적해 우월성질들이지만, 총 작업흐름시간의 최소화를 목적식으로 하는 문제에도 동일하게 적용 될 수 있으며, 그에 대한 증명을 수행하였다. Proposition 3은 Della Croce et al.(2002)의 일반적인 2단계 순열흐름공정과 Choi and Kim[4]의 재투입이 존재하는 2단계 순열흐름공정 일정계획 문제에 관한 연구에서 개발 된 최적해 우월성질들의 기본내용을 바탕으로 개발되었으며 그에 대한 증명을 수행하였다.

Proposition 1(Choi and Kim[4]). 부분스케쥴 σ가 주어진 상태에서, 만약 세 개의 sub-job i^l, k^l, $i^\bullet$($i^l \notin \sigma$, $k^l \notin \sigma$, $i^\bullet \notin \sigma$, $l^\bullet \neq k^2$)가 $C_2(\sigma i^1 k^1) \le C_1(\sigma i^1 k^1 l^\bullet)$를 만족한다면, $\sigma k^1 i^1 l^\bullet$로 시작하는 스케쥴($\sigma k^1 i^1 l^\bullet \pi$)은 $\sigma i^1 k^1 l^\bullet$로 시작하는 스케쥴($\sigma i^1 k^1 l^\bullet \pi$)에 의해 지배된다.

Proof of Proposition 1 : Proposition 1이 본 일정계획 문제의 최적해 우월성질로써 적용 가능함은 다음 (a), (b), (c)의 부등식들이 만족함을 보임으로써 적용가능하다.

(a) $j = 1, 2$에 대해서 $C_j(\sigma i^1 k^1 l^\bullet) \le C_j(\sigma k^1 i^1 l^\bullet)$
(b) $F(\sigma i^1 k^1 l^\bullet) \le F(\sigma k^1 i^1 l^\bullet)$
(c) 스케쥴 π에 의해서 구성된 전체 스케쥴 $\sigma i^1 k^1 l^\bullet \pi$과 $\sigma k^1 i^1 l^\bullet \pi$에 대해서 $c_{k^2 2}(i^1 k^1 l^\bullet \pi) \le c_{k^2 2}(k^1 i^1 l^\bullet \pi)$

(a) $C_j(\sigma i^1k^1l^\bullet) \le C_j(\sigma k^1i^1l^\bullet)$이 만족함을 아래와 같이 우선 증명한다. 우리는 $C_1(\sigma i^1k^1) = C_1(\sigma k^1i^1) = C_1(\sigma^1) + p_{i^11} + p_{k^11}$라는 것을 쉽게 계산할 수 있다. 또한, $l^\bullet \ne k^2$ ($l^\bullet$는 i^2일 수 있음)이기 때문에 아래 수식 (1)을 쉽게 얻을 수 있다.

$$C_1(\sigma i^1k^1l^\bullet) \le C_1(\sigma k^1i^1l^\bullet) \quad (1)$$

또한, 수식 (1)과 Proposition 1의 조건 $C_2(\sigma i^1k^1) \le C_1(\sigma i^1k^1l^\bullet)$로부터 우리는 아래 수식 (2)를 얻을 수 있다.

$$\begin{aligned} C_2(\sigma i^1k^1l^\bullet) &= C_1(\sigma i^1k^1l^\bullet) \\ &+ p_{l^\bullet 2} \le \max\{C_1(\sigma k^1i^1l^\bullet), \\ &C_2(\sigma k^1i^1)\} + p_{l^\bullet 2} = C_2(\sigma k^1i^1l^\bullet) \end{aligned} \quad (2)$$

(b) 수식 (1)과 수식 (2)로부터, $\sigma i^1k^1l^\bullet$에 존재하는 모든 sub-job들은 k^1을 제외하고 $\sigma k^1i^1l^\bullet$에서보다 더 늦게 공정완료 될 수가 없음을 알 수 있다. 또한, job k의 작업흐름시간은 $c_{k^22}(S)$로 정의되기 때문에, 우리는 $F(\sigma i^1k^1l^\bullet) \le F(\sigma k^1i^1l^\bullet)$임을 증명할 수 있다.

(c) 마지막으로, $\sigma i^1k^1l^\bullet$로 시작하는 임의의 스케쥴에서 k^2를 제외하고 모든 second-pass sub-job의 작업시작가능시간(ready time)은 $\sigma k^1i^1l^\bullet$로 시작하는 임의의 스케쥴보다 늦을 수가 없다. 하지만, Proposition 1의 조건$C_2(\sigma i^1k^1) \le C_1(\sigma i^1k^1l^\bullet)$, $l^\bullet \ne k^2$과 수식 (1)과 수식 (2)에 의해서 $\sigma i^1k^1l^\bullet$로 시작하는 k^2의 공정시작가능시간도 $\sigma k^1i^1l^\bullet$로 시작하는 임의의 스케쥴보다 늦을 수 없다는 것을 알 수 있다. 따라서, 임의의 스케쥴 π에 대해서 $c_{k^22}(i^1k^1l^\bullet\pi) \le c_{k^22}(k^1i^1l^\bullet\pi)$이 성립함을 알 수 있다. 이것으로써, Proposition 1이 본 일정계획 문제의 최적해 우월성질로써 적용 가능함을 증명완료 하였다.

Proposition 2(Choi and Kim[4]). 부분스케쥴 σ가 주어진 상태에서, 만약 세개의 sub-job i^2, k^l, $l^\bullet$($i^2 \notin \sigma$, $i^l \in \sigma$ $k^l \notin \sigma$, $i^\bullet \notin \sigma$, $l^\bullet \ne k^2$)가 $C_2(\sigma i^2k^1) \le C_1(\sigma i^2k^1l^\bullet)$, $c_{i^12}(\sigma) \le C_1(\sigma)$를 모두 만족한다면, $\sigma k^1i^2l^\bullet$로 시작하는 스케쥴($\sigma k^1i^2l^\bullet\pi$)은 $\sigma i^2k^1l^\bullet$로 시작하는 스케쥴($\sigma i^2k^1l^\bullet\pi$)에 의해 지배된다.

Proof of Proposition 2. Proposition 2가 본 일정계획 문제의 최적해 우월성질로써 적용 가능함은 Proposition 1의 증명과정과 비슷하게 다음 (a), (b), (c)의 부등식들이 만족함을 보임으로써 적용가능하다.

(a) $j=1, 2$에 대해서 $C_j(\sigma i^2k^1l^\bullet) \le C_j(\sigma k^1i^2l^\bullet)$
(b) $F(\sigma i^2k^1l^\bullet) \le F(\sigma k^1i^2l^\bullet)$
(c) 스케쥴 π에 의해서 구성된 전체 스케쥴 $\sigma i^2k^1l^\bullet\pi$과 $\sigma k^1i^2l^\bullet\pi$에 대해서 $c_{k^22}(i^2k^1l^\bullet\pi) \le c_{k^22}(k^2i^1l^\bullet\pi)$

(a) Proposition 2의 조건들 $c_{i^12}(\sigma) \le C_1(\sigma)$과 $l^\bullet \ne k^2$로부터

$$C_1(\sigma i^2k^1l^\bullet) = C_1(\sigma k^1i^2l^\bullet) \quad (3)$$

또한, 수식 (1)과 Proposition 2의 조건 $C_2(\sigma i^2k^1) \le C_1(\sigma i^2k^1l^\bullet)$로부터 우리는 아래 수식 (2)를 얻을 수 있다.

$$\begin{aligned} C_2(\sigma i^2k^1l^\bullet) &= C_1(\sigma i^2k^1l^\bullet) + p_{l^\bullet 2} \le \max\{C_1(\sigma k^1i^2l^\bullet), \\ &C_2(\sigma k^1i^2)\} + p_{l^\bullet 2} = C_2(\sigma k^1i^2l^\bullet) \end{aligned} \quad (4)$$

(b) 수식 (3)과 수식 (4)로부터, $\sigma i^2k^1l^\bullet$에 존재하는 모든 sub-job들은 k^1을 제외하고 $\sigma k^1i^2l^\bullet$보다 더 늦게 공정완료 될 수가 없음을 알 수 있다. 또한, job k의 작업흐름시간은 $c_{k^22}(S)$로 정의되기 때문에, 우리는 $F(\sigma i^2k^1l^\bullet) \le F(\sigma k^1i^2l^\bullet)$임을 증명할 수 있다.

(c) 마지막으로, $\sigma i^2k^1l^\bullet$로 시작하는 임의의 스케쥴에서 k^2를 제외하고 모든 second-pass sub-job의 공정시작가능시간(ready time)은 $\sigma k^1i^2l^\bullet$로 시작하는 임의의 스케쥴보다 늦을 수가 없다. 하지만, Proposition 2의 조건 $C_2(\sigma i^2k^1) \le C_1(\sigma i^2k^1l^\bullet)$, $l^\bullet \ne k^2$과 수식 (3)과 수식 (4)에 의해서 $\sigma i^2k^1l^\bullet$로 시작하는 k^2의 공정시작가능시간도 $\sigma k^1i^2l^\bullet$로 시작하는 임의의 스케쥴보다 늦을 수 없다는 것을 알 수 있다. 따라서, 임의의 스케쥴 π에 대해서 $c_{k^22}(i^2k^1l^\bullet\pi) \le c_{k^22}(k^1i^2l^\bullet\pi)$이 성립함을 알 수 있다. 이것으로써, Proposition 2이 본 일정계획 문제의 최적해 우월성질로써 적용 가능함을 증명완료 하였다.

Proposition 3 : 부분스케쥴 σ가 주어진 상태에서, 만약 세 개의 sub-job i^2, k^2($i^l \in \sigma$, $i^2 \notin \sigma$, $k^l \in \sigma$, $k^2 \notin \sigma$)가 $j=1, 2$에 대해 $C_j(\sigma i^2k^2) \le C_j(\sigma k^2i^2)$과 $F(\sigma i^2k^2) \le F(\sigma k^2i^2)$를 모두 만족한다면, σk^2i^2로 시작하는 스케쥴($\sigma k^2i^2\pi$)은 σi^2k^2로 시작하는 스케쥴($\sigma i^2k^2\pi$)에 의해 지배된다.

Proof of Proposition 3 : Proposition 3의 조건 $j=1, 2$에 대해 $C_j(\sigma i^2k^2) \le C_j(\sigma k^2i^2)$의해서 σi^2k^2로 시작하는 임의의 스케쥴에서 모든 second-pass sub-job들의 공정완료시간은 σk^2i^2로 시작하는 임의의 스케쥴보다 늦을 수가 없다. 또한, Proposition 3의 조건 $F(\sigma i^2k^2) \le F(\sigma k^2i^2)$이 존재하기 때문에 Proposition 3의 내용은 쉽게 증명된다.

3. 분지한계방법(B&B Algorithm)

본 장에서는 재투입이 존재하는 2단계 순열흐름공정에서 총 작업흐름시간을 최소화하기 위해 개발된 분지한계

방법론을 설명한다. 다음 설명할 분지한계방법론은 다양한 스케쥴링 문제에 대한 전형적인 분지한계방법론[1]을 재투입이 존재하는 2단계 순열흐름공정의 스케쥴링 문제에 적용 가능하도록 개발된 Choi and Kim[2, 4]의 분지한계방법론과 동일하며 자세한 내용은 다음과 같다.

B&B트리는 모든 가능한 스케쥴 혹은 공정순서를 고려한다. B&B트리의 각 노드(node)는 임의의 부분 스케쥴(partial schedule) 혹은 부분 공정순서(partial sequence)를 나타내고, B&B트리의 k번째 레벨에 있는 임의의 노드는 전체 스케쥴 혹은 전체 공정순서의 앞에서부터 k개의 sub-job들에 대한 부분스케쥴을 나타낸다. 본 스케쥴링 문제에서 총 $2n$개의 sub-job들(n개의 first-pass sub-job들과 n개의 second-pass sub-job들)의 공정순서를 고려해야하기 때문에, B&B트리에서 전체 스케쥴 혹은 전체 공정순서를 나타내기 위해서 총 레벨의 숫자는 $2n$개가 된다. 그러나, 본 스케쥴링 문제에서 first-pass sub-job과 그와 관련된 second-pass sub-job의 선후제약조건이 존재하기 때문에, 임의의 first-pass sub-job은 그의 second-pass subjob보다 B&B트리에서 상위레벨에 존재해야 한다.

초기 상한(initial upper bound)값으로는 제 3장에서 제시하는 휴리스틱 해를 사용한다. 분지할 노드의 선택은 깊이우선탐색(depth-first rule)을 사용하고, 만약 동률이 발생한 경우에는 하한값(lower bound)이 가장 작은 노드가 선택된다. 임의의 상위(parent) 노드로부터 하위(child) 노드들이 발생할 경우 하위 노드들에 해당하는 스케쥴들이 앞서 개발한 최적해 우월성질에 의해서 제거되어도 되는지 체크된다. 또한, 제거되지 않은 하위 노드들에 대해서는 아래에서 제시할 총 작업흐름시간에 대한 하한값(lower bound) 계산 방법에 의해서 구해진 하한값과 현 상한(incumbent solution)값을 비교하여, 만약 하한값이 현 상한값과 동일하거나 더 클 경우 해당 노드들은 제거된다. 리프(leaf)노드, 즉, 최하위 레벨의 노드가 발생되면, 그에 대한 총 작업흐름시간을 계산하고 그것이 만약 현 상한 값보다 작다면 현 상한값을 해당 리프노드의 총 작업흐름시간으로 갱신한다.

본 연구에서는 Choi and Kim(2009)의 재투입이 존재하는 2단계 순열흐름공정에서 총 납기지연을 최소를 목적식으로 하는 스케쥴링 문제를 대상으로 하한 값을 계산하기 위해 개발된 성질들을 바탕으로 하한값을 개발한다. 제시된 하한값 계산 방법에서는 고려되고 있는 노드에 관련된 부분 스케쥴에 포함되지 않는 first-pass sub-job과 second-pass sub-job들 사이의 선후제약 조건을 제거(relaxation)하여 하한값을 얻는다[4].

하한값(Lower Bound)

앞서 설명한 바와 같이 본 스케쥴링 문제에서 스케쥴 $S(i^2 \in S)$에서 job i의 흐름시간(작업완료시간)은 설비 2에서 job i의 second-pass sub-job i^2의 작업완료시간으로 정의된다. 따라서, 임의의 부분 스케쥴에서 시작하여 구해진 전체 스케쥴의 총 작업흐름시간에 대한 하한값을 얻기 위해서 우리는 기 수립된 부분 스케쥴에 포함되지 않은 secondpass sub-job들에 대한 흐름시간의 하한값을 구한 뒤 기 수립된 부분 스케쥴에 포함된 second-pass sub-job들의 총 흐름시간에 더해야 한다. 앞서 정의된 기호들에 추가적으로 Choi and Kim[4]의 다음 기호들을 본 연구의 뒷 부분에서 사용한다.

- σ : 분지한계방법론에서 임의의 노드에 해당하는 sub-job들의 부분스케쥴(partial sequence) 혹은 부분 공정순서(partial sequence)
- U : σ에 포함되지 않은, 즉, 아직 스케쥴 되지 않은 sub-job들의 집합
- $p_{[k]j}$: U에 속해있는 sub-job들의 설비 j에서 k번째 짧은 공정시간
- P_{kj} : 집합 U에 속해있는 sub-job들의 설비 j에서 첫 번째부터 k번째 짧은 공정시간들의 합, 즉, $\sum_{q=1}^{k} p_{qj}$
- U'' : U의 sub-job들 중 second-pass sub-job들의 집합
- A'' : σ의 first-pass sub-job들의 해당 second-pass sub-job 들의 집합
- $lb_k^{\cdot}$: U의 sub-job들 중 k번째 공정 완료된 sub-job의 공정완료시간에 대한 하한값
- lb_m^2 : U의 second-pass sub-job들 중 m번째 공정 완료된 second-pass sub-job의 공정완료시간에 대한 하한값

Choi and Kim[4]의 연구에서는 Proposition 4를 이용하여 스케쥴 되지 않은 sub-job들에 대한 각 공정완료시간(흐름시간)에 대한 하한값을 구하고, 그 결과를 바탕으로 Proposition 5를 이용하여 스케쥴 되지 않은 second-pass sub-job들의 각 공정완료시간(흐름시간)에 대한 하한값을 구하였다. 다음, Kim[10]에서 개발 된 성질을 이용하여 총 납기지연에 대한 하한값을 구하였다. 본 스케쥴링 문제의 목적식은 총 작업흐름시간을 최소화하는 것이기 때문에 Choi and Kim[4]의 연구에서 개발 된 아래 Proposition 4와 Proposition 5를 활용하여 스케쥴되지 않은 sub-job들 혹은 second-pass sub-job들의 흐름시간에 대한 하한값을 바로 구할 수 있다.

Proposition 4(Choi and Kim[4]). 재투입이 존재하는 2단계 순열흐름공정에서, σ로 시작하는 임의의 전체 스케

줄에서 U에 속한 sub-job들 중 k번째 완료된 sub-job의 공정완료시간은 $\max\{C_1(\sigma)+P_{k1}+p_{\{1\}2},\ C_2(\sigma)+P_{k2},\ C_1(\sigma)+p_{\{1\}1}+P_{k2}\} \equiv lb_k^{\bullet}$의 값보다 작지 않다. 여기서, $k = 1, 2, \cdots, |U|$.

Proposition 4로부터 σ로 시작하는 임의의 전체 스케줄에서 U에 속한 sub-job들 중 k번째 완료된 sub-job의 공정완료시간에 대한 하한값을 구할 수 있다. 하지만, 본 스케쥴링 문제의 목적식에 대한 하한값을 구하기 위해서는 U에 속한 second-pass sub-job들의 공정완료시간에 대한 하한값을 구해야하며, 이는 다음 Proposition 5를 적용하여 구할 수 있다.

Proposition 5(Choi and Kim[4]). 재투입이 존재하는 2단계 순열흐름공정에서, σ로 시작하는 임의의 전체 스케줄에서 U에 속한 second-pass sub-job들 중 m번째 완료된 second-pass sub-job의 공정완료시간은 $lb_{k'}^{\bullet}$, 즉, $lb_m^2 = lb_{k'}^{\bullet}$. 여기서, 만약 $m = 1, 2, \cdots, |A''|$라면 $k' = 2m - |A''|$이고 $m = |A''|+1, |A''|+2, \cdots, |U''|$.

Proposition 4와 Proposition 5 증명은 Choi and Kim[4]의 연구를 참조하기 바란다.

Proposition 4와 Proposition 5를 이용하여 부분스케줄 σ로 시작하는 임의의 전체스케줄의 총 작업흐름시간에 대한 하한값을 다음과 같은 수식으로 표현할 수 있다.

$$LB(\sigma) = \sum_{i^2} c_{i^2}(\sigma) + \sum_{m=1}^{|U''|} lb_m^2$$

4. 휴리스틱(Heuristic)

제 3장에서는 제안된 분지한계방법에서 사용할 초기 상한값을 구하기 위하여 Choi and Kim[3]의 재투입이 존재하는 순열흐름공정에서 총 작업완료시간을 최소화하기 위해 개발한 휴리스틱 알고리듬을 본 스케쥴링 문제에 적용하였다.

Choi and Kim[3]의 Modified NEH 알고리듬은 재투입이 존재하는 다단계 순열흐름공정에서 총 작업완료시간을 최소화하기 위해 개발된 스케쥴링 방법론을 본 스케쥴링 문제의 목적식(총 작업흐름시간)에 맞게 수정한 것이다. Choi and Kim[3]의 스케쥴링 방법론은 NEH 알고리듬이라고 불리우는 Nawaz et al.[11]의 알고리듬에 기초한다. NEH 알고리듬은 일반적인 다단계 순열흐름공정에서 총 작업완료시간을 최소화하기 위해 개발된 알고리듬이다. NEH 알고리듬을 본 연구의 스케쥴링 문제에 적용 가능하도록 수정한 알고리듬이다. NEH 알고리듬에서 job들은 모든 설비에서 공정시간들의 합한 값에 따라 내림차순으로 정렬되고, 각 step마다 정렬된 job들을 순차적으로 선택하여 부분스케쥴의 목적식이 가장 작은 값을 갖도록 최적위치에 할당함으로써 마지막 step에서 최종해(sequence)를 얻는방법이다. 즉, k번째 step에서는 정렬된 job들 중 k번째 job을 선택하여 k-1번째 step에서 k-1개의 job들로 구해진 부분스케쥴(partial sequence)에 존재하는 k개의 대안(alternative posisition)들 중 부분스케쥴의 목적식을 최소화하는 최적위치에 할당되는 것이다.

Choi and Kim[3]의 알고리듬에서는, 각 step마다 선택된 sub-job의 최적위치를 선택하기 위해, first-pass sub-job과 second-pass sub-job의 선후제약조건을 만족하는 위치를 대안으로 한정시켰다. 또한, 각 step에서 선택된 sub-job이 삽입될 위치, 즉, 각 대안들을 평가할 때 부분스케쥴에 포함되지 않은 sub-job들은 현 부분스케쥴 뒤에 스케쥴된다는 가정을 하여 각 대안에 해당하는 부분스케쥴의 목적식이 아닌 전체스케쥴의 목적식을 평가하여 최적위치를 선택하였다. 여기서, Choi and Kim[3]의 알고리듬에서는 각 대안에 대한 평가가 총 작업완료시간으로 이루어졌지만, 본 연구에서는 총 작업흐름시간으로 이루어진다. 위와 같은 일련의 과정은 전진 및 후진의 양방향으로 스케쥴이 더 이상 향상되지 않을 때까지 반복적으로 이루어진다.

Choi and Kim[3]의 연구결과를 보면 본 MN 알고리듬은 기존의 디스패칭 규칙이나 NEH 알고리듬 보다 훨씬 좋은 결과를 제시하고 있다. 보다 자세하고 정확한 MN 알고리듬의 프로시져를 설명하기 위해 다음과 같이 정리하였다.

Procedure of MN

Step 1 : sub-job들의 총 공정시간($\sum_{j=1}^{2} p_{i^k j}$)에 따라 오름차순으로 정렬된 집합(sorted sequence)을 구성한다.

Step 2 : 정렬된 집합에서 first-pass sub-job과 second-pass sub-job의 선후제약조건을 만족하지 않는 second-pass sub-job들을 선택하여 각 해당 first-pass sub-job의 뒤에 재 할당된 집합을 S^0로 정의한다.

Step 3 : $u = 1$에서 $u = 2n$까지 다음(3.1-3.3)을 반복 수행하여 얻은 집합을 S^N로 정의한다.

3.1) S^0에서 u번째 sub-job을 선택한다.

3.2) $k = 1$에서 $k = |\sigma|$까지 다음을 수행한다: 부분스케쥴 σ에서 (k-1)번째 sub-job과 k번째 sub-job사이의 위치들 중 first-pass sub-job과 secondpass sub-job의 선후제약조건을 만족(feasible)

하는 위치들에 대해서 각각 전체스케줄 S의 총 작업흐름시간을 계산한다. 여기서, S는 $|\sigma|+1$개의 sub-job들의 부분스케쥴 뒤에 S^0의 $u+1$번째부터 $2n$번째 위치에 있는 sub-job들이 차례로 스케쥴 된 전체스케쥴을 의미한다.

3.3) Step 3.3에서 목적식의 최소값을 제공하는 위치에 선택된 sub-job을 할당하여 얻어진 부분스케쥴을 σ로 업데이트한다.

Step 4 : 만약 Step 3에서 얻어진 S^N의 총 작업흐름시간이 S^0의 총 작업흐름시간보다 작지 않다면, 본 알고리듬의 수행을 종료하고, 그렇지 않다면, S^0를 S^N으로 리셋($S^0=S^N$)후 Step 5로 간다.

Step 5 : $u=2n$에서 $u=1$까지 다음(3.1-3.3)을 반복 수행하여 새로운 S^N을 얻는다.

Step 4. 만약 Step 5에서 얻어진 S^N의 총 작업흐름시간이 S^0의 총 작업흐름시간보다 작지 않다면, 본 알고리듬의 수행을 종료하고, 그렇지 않다면, S^0를 S^N으로 리셋($S^0=S^N$)후 Step 3로 간다.

5. 실험(Computational Experiments)

본 연구에서 제안한 방법론들의 성능 평가를 위해 제안된 스케쥴링 알고리듬들은 C언어로 코딩 후 2.20GHz 클럭 속도에서 운영되는 펜티엄 듀얼 CPU를 탑재한 개인용 퍼스널 컴퓨터를 이용하여 실험을 수행하였으며 분지한계방법론들은 모두 최적해를 제공하기 때문에 그 성능들을 수행시간에 초점을 맞추어 비교하였다.

우선, 앞서 제안된 하한값의 효율성을 보기 위해 제안된 하한값을 보유한 분지한계방법론과 보유하지 않은 분지한계방법론의 성능을 비교하였다. 본 실험을 위해, sub-job들의 개수를 4개 레벨(10, 12, 14, 16)과 설비에서 sub-job들의 공정시간범위와 관련된 두 개의 레벨($R=1$, 10)의 각 조합마다 5개의 문제를 무작위 발생시켜 총 40개의 문제를 통해 실험을 수행하였다. 여기서, sub-job들의 각 설비에서의 공정시간은 [1, 10R]의 범위를 갖는 이산균등분포(discrete uniform distribution)에서 무작위 발생시켰다. 또한, 실험에서 지나친 시간의 소요를 방지하기 위해, 분지한계방법론들은 1800초 내에 최적해를 제공하지 못할 경우 강제종료를 하도록 실험하였다.

본 실험의 결과값은 <Table 1>에서 두 개의 분지한계방법론들의 수행시간과 발생된 노드(node)들의 비율로 확인할 수 있다. <Table 1>에서 BB1과 BB2는 하한값을 보유한 분지한계방법론과 보유하지 않은 분지한계방법론을 의미하며, BB1과 BB2에는 앞서 제안된 3개의 최적해 우월성질(dominance property)들과 MN에 의해서 구해진 초기 상한값이 모두 구현되어 있는 상태이다. 본 실험의 결과를 보면 수행시간과 발생된 노드수 측면에서 BB1이 BB2에 비해 월등히 좋은 성능을 보여주고 있다. 즉, BB1에서 사용된 하한값이 매우 효율적이라는 것을 의미한다.

<Table 1> 하한값의 효율성에 대한 실험결과

sub-job 수	ACPUT†		ARCPUT‡	ARSC#
	BB1	BB2	(BB1/BB2)	(BB1/BB2)
10	0.001	0.05	2.0×10^{-1}	6.2×10^{-3}
12	0.004	2.39	2.2×10^{-3}	7.1×10^{-4}
14	0.099	147	9.4×10^{-4}	1.9×10^{-4}
16	1.192	1800	-	-

† 평균수행시간 또는 수행시간에 대한 하한값(1800초 내에 최적해를 구하지 못할 경우 수행시간을 1800초로 가정함)
‡ 수행시간 비율의 평균
# 발생된 노드수 비율의 평균

두 번째와 세 번째 실험들은 앞서 제안된 최적해 우월성질들과 초기해를 구하기 위한 MN 휴리스틱 방법론의 효율성을 보기 위해 제안된 우월성질들을 보유한 분지한계방법론(BB1)과 보유하지 않은 분지한계방법론(BB3), 그리고 MN 휴리스틱 방법론으로 초기 상한값을 구한 분지한계방법론(BB1)과 MN 휴리스틱 방법론에서 Step 1, 2만을 이용하여 초기 상한값을 구한 분지한계방법론(BB4)의 성능을 비교하였다. 물론, BB3는 본 연구에서 제안된 하한값과 MN에 의해 구해진 초기 상한값을 사용하고, BB4는 하한값과 우월성질들을 사용한다. 본 실험을 위해, sub-job들의 개수를 4개 레벨(14, 16, 18, 20)과 설비에서 sub-job들의 공정시간범위와 관련된 두 개의 레벨($R=1$, 10)의 각 조합마다 5개의 문제를 무작위 발생시켜 총 40개의 문제를 통해 실험을 수행하였다.

본 실험들의 결과값은 <Table 2>, <Table 3>에서 확인할 수 있다. <Table 1>의 결과와 마찬가지로 <Table 2>의 결과를 보면 BB1에서 사용된 우월성질들이 매우 효율적이라는 것을 의미한다. 또한, <Table 3>의 결과를 보면

<Table 2> 최적해 성질들의 효율성에 대한 실험결과

sub-job 수	ACPUT†		ARCPUT‡	ARSC#
	BB1	BB3	(BB1/BB3)	(BB1/BB3)
14	0.099	0.163	0.729	0.650
16	1.192	1.934	0.644	0.605
18	6.271	10.52	0.600	0.575
20	100.1	224.1	0.532	0.519

† ‡ # <Table 1> 참고바람.

발생된 노드들의 수를 보면 BB1이 BB4 보다 효율적인 것 같지만 수행시간 측면에서는 BB1과 BB4의 성능에는 큰 차이가 없는 것으로 보인다. 이는 MN에 의해 구해진 초기 상한값에 의해 분지한계방법론에서 노드의 수를 어느 정도 줄일 수 있지만, MN 방법론 자체의 구동시간이 걸리기 때문일 것으로 추측된다.

<Table 3> MN알고리듬의 효율성에 대한 실험결과

sub-job 수	ACPUT†		ARCPUT‡	ARSC#
	BB1	BB4	(BB1/BB4)	(BB1/BB4)
14	0.099	0.097	1.010	0.980
16	1.192	1.177	1.000	0.978
18	6.271	6.291	0.995	0.991
20	100.1	102.6	0.964	0.985

† ‡ # <Table 1> 참고바람.

마지막 <Table>은 BB1의 성능을 평가하기 위한 본 연구의 주요 실험이다. 즉, 본 연구에서 제안된 하한값, 우월성질, 초기 상한값을 보유한 BB1의 성능을 평가하는 것이 목적이다. 본 실험을 위해, sub-job들의 개수를 6개 레벨(14, 16, 18, 20, 22, 24)과 설비에서 sub-job들의 공정시간범위와 관련된 두 개의 레벨($R=1$, 10)의 각 조합마다 5개의 문제를 무작위 발생시켜 총 60개의 문제를 통해 실험을 수행하였다.

본 실험들의 결과 값은 <Table 4>에서 확인할 수 있다. <Table 4>의 결과를 보면 sub-job들의 수가 22개 까지는 적절한 수행시간 내에 대부분의 문제에서 최적해를 줄 수 있다는 것을 확인할 수 있다.

<Table 4> BB1 알고리듬의 주요 실험결과

sub-job 수	R	ACPUT†	MCPUT‡	NPNS#
14	1	0.1	0.1	0
	10	0.1	0.1	0
16	1	0.7	0.5	0
	10	1.7	0.9	0
18	1	5.8	4.6	0
	10	6.7	5.8	0
20	1	48.3	24.6	0
	10	152.1	86.6	0
22	1	576.0	327.1	0
	10	842.5	362.5	1
24	1	1200	1173	2
	10	1650	1800	4

† <Table 1> 참조.
‡ 수행시간의 중간값.
# 5문제 중 1800초 내에 최적해를 구하지 못한 문제의 수.

6. 결 론

본 연구에서는 재투입이 있는 2단계 순열흐름공정에서 총 작업흐름시간을 고려하는 스케쥴링 문제를 다루고 있다. 재투입이 있는 흐름공정이란 일반적인 흐름공정의 확장된 형태이다. 일반적인 흐름공정에서는 제품이 순차적으로 위치하고 있는 각 기계들에서의 해당 공정을 통하여 이루어지며 마지막 기계에서의 해당 공정이 완료될 때 그 제품의 작업이 완료 된다. 하지만, 재투입이 있는 흐름공정에서는 제품이 마지막 기계에서의 해당공정을 완료한 후 흐름공정의 첫 번째 기계에 재투입되어 처음에 수행했던 공정과 유사한 혹은 동일한 공정들을 다시 수행하게 된다. 재투입이 있는 흐름공정은 반도체 웨이퍼와 PCB(printed circuit board) 등의 high-tech 제품 및 섬유와 거울을 생산하는 제조 시스템에서 빈번히 사용되는 공정이며 일반적으로 병목공정으로 분류된다.

재투입이 존재하는 2단계 순열흐름공정 문제의 최적해를 구하기 위해 본 연구에서는 분지한계방법론을 제안하였고, 분지한계방법론의 효율성을 높이기 위해 기존 연구에서 다른 목적식을 위해 개발된 최적해 우월성질들이 본 문제에 적용 가능함을 증명한 뒤 사용하였고 새로운 최적해 우월성질을 개발하였다. 또한 하한값, 휴리스틱 방법론들을 제안하였다. 실험결과 22개의 sub-job까지는 본 연구에서 개발한 분지한계방법론으로 적당한 시간 내 최적해를 구할 수 있었으며, 실험결과 하한값과 최적해 우월성질들은 매우 효율적인 것으로 판단된다.

본 연구는 각 공정에서 한 대의 설비가 존재한다고 가정하였지만, 실제 제조라인에서는 각 공정의 작업을 수행하는 병렬설비가 존재할 수 있고, 작업준비교체시간 및 버퍼의 제약이 존재할 수 있어 바로 현장에 적용하기에 어려움이 있을 수 있다. 따라서, 추후 연구에서는 이러한 현실적인 제약조건 및 현장의 실제 데이터를 반영한 일정계획 문제에 대한 연구를 수행할 계획이다.

참고문헌

[1] Baker, K. R.; Introduction to Sequencing and Scheduling. John Wiley and Sons : New York, 148-152, 1974.

[2] Choi, S.-W. and Kim, Y.-D.; "Minimizing makespan on a two-machine re-entrant flowshop," *Journal of the Operational Research Society*, 58(7) : 972-981, 2007.

[3] Choi, S.-W. and Kim, Y.-D.; "Minimizing makespan on an m-machine re-entrant flowshop," *Computers and Operations Research*, 35(5) : 1684-1696, 2008.

[4] Choi, S.-W. and Kim, Y.-D.; "Minimizing total tardiness on a two-machine re-entrant flowshop," *Journal of the Operational Research Society*, 199(2) : 375-384, 2009.

[5] Della Croce, F., Narayan, V., and Tadei, R.; "The two-machine total completion time flow shop problem," *European Journal of Operational Research*, 90(2) : 227-237, 1996.

[6] Demirkol, E. and Uzsoy, R.; "Decomposition methods for reentrant flow shops with sequence dependent setup times," *Journal of Scheduling*, 3(3) : 115-177, 2000.

[7] Graves, S. C., Meal, H. C., Stefek, D., and Zeghmi, A. H.; "Scheduling of re-entrant flow shops," *Journal of Operations Management*, 3(4) : 197-207, 1983.

[8] Hoogeveen, J. A. and van de Velde, S. L.; "Stronger Lagrangean bounds by use of slack variables : Application to machine scheduling problems," *Mathematical Programming*, 70(1-3) : 173-190, 1995.

[9] Ignall, E. and Schrage, L. E.; "Application of the branch-andbound technique to some flow shop problems," *Operations Research*, 13(3) : 400-412, 1965.

[10] Kim, Y.-D.; "A new branch and bound algorithm for minimizing mean tardiness in two-machine flowshops," *Computers and Operations Research*, 20(4) : 391-401, 1993.

[11] Nawaz, M., Enscore, E. E., and Ham, I.; "A heuristic algorithm for the m-machine, n-job flow-shop scheduling problem," *Omega*, 11(1) : 91-95, 1983.

[12] Pan, J. C. H. and Chen, J. S.; "Minimizing makespan in re-entrant permutation flow-shops," *Journal of Operational Research Society*, 54(6) : 642-653, 2003.

[13] van de Velde, S. L.; "Minimizing the sum of the job completion times in the two-machine flow shop by Lagrangean relaxation," *Annals of Operations Research*, 26(1) : 257-268, 1990.