

# 작업순서 의존형 준비시간을 갖는 이중병렬기계의 휴리스틱 일정계획

고시근<sup>†</sup> · Karunia A. Mahardini

부경대학교 시스템경영공학부

## Heuristics for Non-Identical Parallel Machine Scheduling with Sequence Dependent Setup Times

Shiegheun Koh · Karunia A. Mahardini

Department of Systems Management and Engineering, Pukyong National University

This research deals with a problem that minimizes makespan in a non-identical parallel machine system with sequence and machine dependent setup times and machine dependent processing times. We first present a new mixed integer programming formulation for the problem, and using this formulation, one can easily find optimal solutions for small problems. However, since the problem is NP-hard and the size of a real problem is large, we propose four heuristic algorithms including genetic algorithm based heuristics to solve the practical big-size problems in a reasonable computational time. To assess the performance of the algorithms, we conduct a computational experiment, from which we found the heuristic algorithms show different performances as the problem characteristics are changed and the simple heuristics show better performances than genetic algorithm based heuristics for the case when the numbers of jobs and/or machines are large.

**Keywords:** Scheduling, Non-Identical Parallel Machine, Sequence Dependent Setup, Heuristic

### 1. 서론

일반적으로 대규모 제조기업에서는 비교적 복잡한 구조의 제품을 생산하기 위해 여러 단계의 공정으로 이루어진 생산시스템을 운영하는데, 이 경우 제품들은 각 단계의 공정을 거치며 점진적으로 완제품의 형태를 갖추어간다. 반면, 중소 제조기업의 경우에는 다른 제조기업(원청회사)으로부터 주문을 받아 비교적 간단한 구조의 전문화된 제품을 생산하는 경우가 많은데, 본 연구에서는 이러한 형태의 중소규모 제조기업에서 발생하는 문제를 해결하고자 한다. 이 기업의 핵심 제조시스템은 <Figure 1>과 같이 유사한 기계들의 그룹으로 이루어진 한 단계의 공정으로 이루어져 있는데, 여러 고객사들로부터 유사한 종류의 제품에 대한 주문을 받아 이 시스템에서 처리하고 있다.

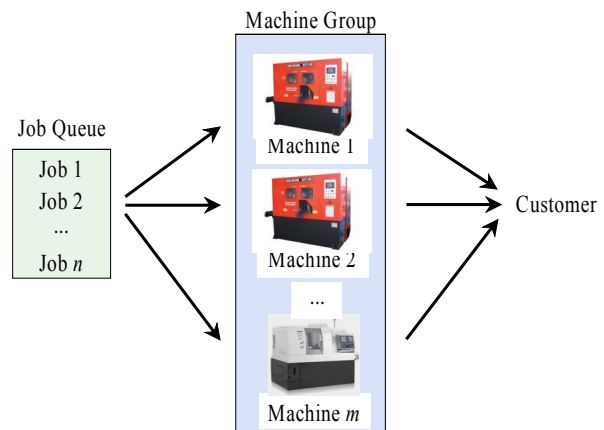


Figure 1. Non-identical parallel machine system

이 논문은 2011학년도 부경대학교의 지원을 받아 수행된 연구임(PK-2011-0478).

<sup>†</sup> 연락저자 : 고시근 교수, 608-737 부산광역시 남구 용소로 45, 부경대학교 시스템경영공학부, Tel : 051-629-6484, Fax : 051-629-6478,

E-mail : sgkoh@pknu.ac.kr

2013년 9월 25일 접수; 2014년 1월 9일 수정본 접수; 2014년 2월 12일 게재 확정.

각 주문은 그룹에 속한 기계들 중 하나를 통해 처리되는데, 이 기계들은 일반적으로 기업이 한꺼번에 일괄 구입한 것이 아니고 생산시스템을 운영하면서 필요에 따라 한 대씩 구매하였기 때문에 용도는 유사하지만 그 성능은 조금씩 차이가 나는 것이 보통이다. 이렇게 용도는 같지만 성능 및 사양이 서로 다른 기계그룹을 ‘이종병렬기계(Non-identical Parallel Machine)’ 그룹이라고 부른다. 이러한 이종병렬기계 생산시스템은 플라스틱 사출이나 기계 가공 등 다양한 분야의 기업에서 쉽게 찾아볼 수 있다.

이러한 시스템에서 처리되는 주문들을 살펴보면 대체로 지속형과 일회성의 두 가지 형태로 분류할 수 있다. 일회성 주문은 글자 그대로 반복성이 없는 주문인데 반해 지속형 주문은 장기간에 걸쳐 지속적으로 들어오는 주문으로서 하루에 일정량씩 매일(혹은 자주) 납품해야 하는 형태를 갖는다. 이 경우 하루 납품량이 기계의 하루 생산능력을 초과하면 한 대 이상의 기계가 그 제품의 생산을 전담해야겠지만(전담기계의 경우는 본 연구의 고려대상이 아님) 일반적으로는 하루 납품량이 기계 생산능력보다 작아서 다른 일회성 혹은 지속형 주문들과 기계를 공유하는 것이 보통이다. 이러한 지속형 주문은 일자별 납품량을 하나의 주문으로 취급하면 납기가 서로 다른 여러 개의 주문으로 분리해서 생각할 수 있다. 물론 이 분리된 주문들은 원래 한 종류의 제품에 대한 주문이었으므로 이 주문들 사이에는 기계준비시간(셋업시간)이 필요 없을 것이다.

이렇게 다양한 형태의 주문들(여러 개로 나누어진 지속형 주문들과 일회성 주문들)이 이종병렬기계 시스템 앞에서 처리를 기다리고 있을 때, 각 주문들을 어느 기계에 할당할 것인지 그리고 할당된 기계 안에서 처리되는 순서는 어떻게 되는지를 결정하는 것이 본 연구의 문제이다. 이 문제의 해결에 있어 중요한 변수는 기계의 성능차이와 주문에 대한 기계준비시간이다. 즉, 기계의 성능차이로 인해 각 주문을 어느 기계에 할당하느냐에 따라 그 주문의 처리시간이 달라지게 되므로 주문의 기계할당 문제가 중요해지고, 또한 주문의 처리순서에 따라 기계준비시간이 달라지기 때문에 주문의 처리순서 결정문제도 중요해지는 것이다.

일반적으로 제조시스템에서 가장 중요한 관리목표는 생산성의 제고일 것이다. 따라서 본 연구에서도 대상 시스템의 목표를 생산성 제고에 맞추기로 한다. 그런데 일반적으로 기계의 생산성을 제고하기 위해서는 그 가동률을 제고하는 것이 필요하며, 또한 가동률의 제고를 위해서는 작업 전체완료시간(makespan)의 최소화가 필요하다. 따라서 본 연구의 목적은 효과적인 일정계획을 통해 작업 총 완료시간을 최소화(Minimize the makespan)하는 것으로 한다.

본 연구의 나머지 부분은 다음과 같이 구성된다. 다음 절에서는 이종병렬기계 일정계획에 관련된 선행연구들을 조사한다. 제 3장에서는 대상문제를 정수계획법의 형태로 모형화하고, 제 4장에서는 대상문제를 해결하기 위한 비교적 단순한 형태의 휴리스틱 알고리즘을 제안한다. 제 5장에서는 유전 알고

리즘 기반의 해법을 제시하며, 제 6장에서는 제안된 해법들의 성능을 평가하기 위한 수치실험을 소개한다. 마지막으로 제 7장에서는 논문을 요약하고 추후 연구과제를 제시한다.

## 2. 연구 동향

국내외적으로 생산설비의 일정계획을 다룬 연구는 1950년대 중반부터 그 수를 셀 수 없을 정도로 많이 수행되었다. Allahverdi *et al.*(2008)에 의하면 수천 건의 관련 문헌을 찾을 수 있으며, 이 연구들의 대부분은 셋업시간이 무시할 수 있을 정도로 작거나 작업시간에 포함될 수 있는 형태라고 가정하였다. 셋업시간을 작업시간과 별도로 고려한 연구는 1960년대 중반부터 수행되기 시작하였으며, 관련 연구결과들은 Allahverdi *et al.*(1999, 2008)과 Potts and Kovalyov(2000)와 같은 서베이 논문에 잘 정리되어 있다. 이 서베이 논문들에는 셋업시간이 작업순서에 무관 및 유관한 상황, 주문들이 묶음(batch)으로 처리되거나 혹은 그렇지 않은 상황, 다양한 기계 상황(기계 한 대, 병렬 기계, 흐름 라인 등)에 대해 기존 연구문헌들을 체계적으로 정리하였다. 특히 Cheng and Sin(1990)은 병렬기계를 다룬 기존 연구문헌들만 별도로 조사하기도 하였다.

일반적인 생산일정 계획문제는 이와 같이 많이 연구되었지만 본 연구의 상황인 이종병렬기계 시스템에서 작업순서에 의존하는 셋업시간이 존재하는 경우를 다룬 연구는 그리 많지 않다. 이러한 상황을 다룬 연구는 Marsh and Montgomery(1973)가 최초로 생각되는데, 그들은 동일기계 혹은 이종기계 병렬 시스템에서 셋업시간을 최소화하는 휴리스틱 방법을 제안하였다. 그 후 Guinet(1990)은 순서의존적인 셋업시간을 갖는 이종병렬기계 시스템에서 평균흐름시간을 최소화하기 위한 혼합정수계획 모형을 제시하고 모형을 풀기 위한 휴리스틱 알고리즘을 개발하였다. Zhu and Heady(2000)는 비슷한 상황에서 조기완료 및 지연완료를 최소화하는 혼합정수계획 모형을 개발하였으며 그 모형을 사용해 기계가 3대이고 주문이 9개인 문제의 최적해를 적정시간 안에 구하였다. Weng *et al.*(2001)은 비슷한 상황에서 각 작업들의 완료시간 가중합을 최소화하는 문제를 해결하기 위해 7개의 간단한 휴리스틱을 제안하였으며 수치실험을 통해 그 휴리스틱들을 평가하였다.

국내에서는 Jeon and Kang(1995)이 셋업시간에 대한 고려 없이 이종병렬기계 시스템의 납기 지연 작업수를 최소화하기 위한 Tabu Search 방법론을 제안하였다. Kang *et al.*(1998)도 셋업시간을 고려하지 않은 이종병렬기계 시스템에서 각 주문들이 서로 다른 납기를 가질 때 납기 지연 주문의 수를 최소화하는 알고리즘을 개발하였다.

비교적 최근 들어 관련연구의 빈도가 높아지고 있는데, Hop and Nagaur(2004)는 PCB 생산라인의 이종병렬기계 상황에서 의존하는 준비시간을 포함한 총 완료시간을 최소화하기 위하여 수리적 모형을 제시한 다음 유전 알고리즘 기반의

해법을 제안하였다. Agarwal *et al.*(2006)은 이중병렬기계 시스템의 총 완료시간(makespan)을 최소화하기 위해 12개의 휴리스틱과 augmented neural network 방법론을 제안하고 이 해법들을 비교·평가하였다. Li and Yang(2009)은 이중병렬기계 상황을 다룬 연구들 중에서 총 가중 완료시간의 최소화를 목적으로 하는(즉, 납기를 고려하지 않은) 연구들을 수집하고 해법에 따라 분류한 결과를 소개하였다. Tavakkoli-Moghaddam *et al.*(2009)은 순서 및 기계에 의존하는 셋업시간을 고려한 이중병렬기계 시스템에서 지연작업의 수와 총 완료시간을 최소화하기 위한 유전 알고리즘 기반의 해법을 제안하였다. Gharegozli *et al.*(2009)은 순서에 의존하나 기계와는 무관한 셋업시간을 가정하고 총 가중흐름시간과 총 가중지연시간을 동시에 최소화하기 위한 혼합정수 목표계획 모형을 개발하였다. 이 두 연구에서는 공통적으로 주문들은 각각의 도착시간을 갖고 있어서 이 시간 이전에는 작업이 진행될 수 없음을 가정하였다. Vallada and Ruiz (2011)은 순서 및 기계 의존적인 셋업시간을 고려한 문제를 해결하기 위해 지역탐색 알고리즘과 유전 알고리즘을 결합한 알고리즘을 개발하였다. 가장 최근에는 Joo and Kim(2012)이 순서 및 기계에 의존하는 기계준비시간을 고려한 이중병렬기계 일정계획문제를 다루었다. 총 완료시간을 최소화하는 수리 모형을 제시하고 유전 알고리즘 및 자기진화(self-evolution) 알고리즘 기반의 두 해법을 개발하고 그 성능을 비교·평가하였다.

본 연구의 대상문제가 갖는 가장 중요한 특성은 1)이 중병렬기계, 2) 순서에 따라 변하는 셋업시간, 3) 기계에 따라 변하는 셋업시간, 4) 기계 가용시간, 5) 주문 도착시간 등 5개로 요약할 수 있는데 기존의 연구들 중 본 연구과제와 동일한 상황을 다룬 연구는 아직 없었다. 특히 4번째 항목은 현실적으로는 자주 등장하는 상황인데 지금까지 다루어진 적이 한 번도 없었다. 이 항목은 계획시점 현재 그룹 내 일부 기계는 처리중인 작업이 있어서 다른 주문을 바로 처리할 수 없다는 것을 의미한다. 실제로 본 연구에서 다루는 일정계획 문제가 발생하는 시점은 하나의 기계에서 생산이 종료되었을 때이며, 계획대상 주문들은 그 시점 이전에 접수되어 있거나 미래의 어떤 시점에 접수되기로 예약되어 있는 주문들이다. 따라서 4번째 항목과 같이 계획시점에 각 기계들은 처리중인 주문을 가질 수 있으며 그 완료시간(즉, 각 기계의 가용시간)은 모두 다른 값을 가지게 되는 것이다. 본 연구에서는 이러한 상황에서 총 완료시간의 최소화를 목적함수로 하는 최적화 문제를 해결하고자 한다.

### 3. 최적화 모형

본 장에서는 앞 절에서 소개한 문제의 수리적 모형을 개발한다. 본 연구에서 다루는 문제는 Zhu and Heady(2000) 및 Joo and Kim(2012)의 문제와 매우 유사하므로 수리적 모형에 있어 큰 차이는 없다. 우선 모형에서 사용할 기호들을 먼저 설명한 다음 최적화 모형을 제시한다.

<파라미터>

$i, j$  : 주문을 표시하는 인덱스( $i, j = 1, 2, \dots, n$ )

$k$  : 기계를 표시하는 인덱스( $k = 1, 2, \dots, m$ )

$a_i$  : 계획시점 이전에 주문  $i$ 가 도착하였으면 0, 그렇지 않으면 도착예정시간

$r_k$  : 계획시점에 기계  $k$ 가 작업중이면 그 작업의 완료시간, 그렇지 않으면 0

$s_{ik}$  : 기계  $k$ 에서 가장 먼저 처리되는 주문이  $i$ 일 경우의 셋업시간

$s_{ijk}$  : 기계  $k$ 에서 주문  $i$ 가 완료되고 주문  $j$ 를 준비할 경우의 셋업시간

$p_{ik}$  : 주문  $i$ 가 기계  $k$ 에 할당되었을 경우의 처리시간(셋업시간 제외)

<결정변수>

$x_i$  = 주문  $i$ 의 착수시간(셋업시간이 0보다 큰 경우에는 셋업 착수시간)

$f_i$  = 주문  $i$ 의 완료시간

$f_{\max}$  = 모든 주문의 완료시간(makespan)

$y_{ik}$  = 주문  $i$ 가 기계  $k$ 에 할당되면 1, 그렇지 않으면 0

$z_{ik}$  = 기계  $k$ 에서 가장 먼저 처리되는 주문이  $i$ 이면 1, 그렇지 않으면 0

$z_{ijk}$  = 기계  $k$ 에서 주문  $i$ 에 이어 주문  $j$ 가 처리되면 1, 그렇지 않으면 0

<모형>

$$\text{Minimize } f_{\max} \tag{1}$$

subject to

$$x_i \geq a_i \text{ for all } i \tag{2}$$

$$x_i + M(1 - z_{ik}) \geq r_k \text{ for all } i, k \tag{3}$$

$$x_i + \sum_{k=1}^m (s_{ik}z_{ik} + \sum_{\substack{j=1 \\ j \neq i}}^m s_{jik}z_{jik} + p_{ik}y_{ik}) = f_i \text{ for all } i \tag{4}$$

$$x_j + M(1 - \sum_{k=1}^m z_{ijk}) \geq f_i \text{ for all } i, j \tag{5}$$

$$f_i \leq f_{\max} \text{ for all } i \tag{6}$$

$$\sum_{k=1}^m y_{ik} = 1 \text{ for all } i \tag{7}$$

$$\sum_{i=1}^n z_{ik} = 1 \text{ for all } k \tag{8}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^n z_{ijk} \leq y_{ik} \text{ for all } i, k \tag{9}$$

$$z_{ik} + \sum_{\substack{j=1 \\ j \neq i}}^n z_{jik} = y_{ik} \text{ for all } i, k \tag{10}$$

$$y_{ik}, z_{ik} : 0/1 \text{ integer for all } i, k$$

$$z_{ijk} : 0/1 \text{ integer for all } i, j, k$$

식 (1)은 본 문제가 총 완료시간을 최소화하는 문제라는 것을 보여준다. 식 (2)는 각 주문이 도착한(확정된) 이후에 작업이 시작될 수 있음을 나타낸다. 식 (3)은 각 기계가 현재 작업 중인 작업을 종료한 이후에 새로운 작업을 시작할 수 있다는 조건을 표현하고 있다. 이 조건은 매우 현실적인 조건이나 이러한 형태의 제약조건을 포함한 모형은 기존의 연구에서 다루어지지 않았다. 식 (4)는 각 주문의 착수시간과 완료시간 사이의 관계를 나타낸다. 식 (5)에서는 각 기계에서 선행 주문의 작업이 완료되어야만 후속 주문의 작업이 시작될 수 있음을 표현하고 있다. 식 (6)은 총 완료시간이 모든 주문들의 완료시간 중 가장 큰 값이라는 것을 보여준다. 식 (7)은 각 주문이 한 대의 기계에만 할당되도록 해준다. 식 (8)은 각 기계에 최초로 배정되는 작업은 한 개라는 것을 보여준다. 식 (9)에 의하면 어떤 주문이 어떤 기계에 배정되지 않았다면( $y_{ik} = 0$ ) 그 주문은 그 기계에서 다른 어떤 주문의 선행주문도 될 수 없다( $z_{ijk} = 0$ )는 것을 보여준다. 반대로 해석하면 어떤 주문이 하나의 기계에 배정되었다면( $y_{ik} = 1$ ) 그 주문은 그 기계에서 다른 어떤 주문의 선행작업이라는 것( $z_{ijk} = 1$ )을 보여준다. 그런데 이 식이 부등식인 것은 그 주문이 그 기계에서 최종으로 처리되는 주문일 경우 때문이다. 비슷하게 식 (10)에서는 어떤 주문이 하나의 기계에 배정되었다면( $y_{ik} = 1$ ) 그 주문은 그 기계에서 처리되는 최초의 주문( $z_{ik} = 1$ )이거나 다른 어떤 주문의 후속주문이라는 것( $z_{jik} = 1$ )을 보여준다.

이 모형은 문제의 크기(주문수 및 기계수)가 작은 경우 CPLEX 나 LINGO 같은 최적화 소프트웨어를 사용하여 쉽게 풀 수 있다. 그러나 잘 알려진 바처럼 이 문제는 NP-hard에 속하므로 문제의 크기가 커질 경우 정상적인 시간 안에 최적해를 구하는 것은 불가능한 일이다. 실제로 저자는 LINGO를 사용해 주문의 수가 8이고 기계수가 3인 상황의 최적해를 1분 안에 구할 수 있었으나 그 이상의 크기를 갖는 문제는 1시간 안에 해를 구할 수 없었다. 따라서 본 연구에서는 최적은 아니지만 정상적인 시간 안에 최적에 가까운 좋은 해를 찾을 수 있는 휴리스틱 알고리즘을 개발하는데 초점을 맞추도록 한다.

#### 4. 단순 휴리스틱 방법론

앞에서 제시한 모형을 해결하기 위해 본 연구에서는 두 개의 단순 휴리스틱과 두 개의 유전 알고리즘 기반 알고리즘을 개발하였다. 첫 번째 휴리스틱은 BH(Base Heuristic) 라고 명명하였으며 그 수행절차는 다음과 같다.

##### BH(Base Heuristic) Algorithm

Step 0 :  $N =$  계획대상 주문의 집합,  $M =$  전체 기계 집합

Step 1 : 다음과 같은 방법으로 배치대상 기계를 선택한다.

- 1.1)  $M$ 에 포함된 모든 기계에 대해  $F_k (=$  기계  $k$ 에 배치된 작업들이 모두 완료되는 시점)를 계산한다. 배치된 작업이 없

다면  $F_k = r_k$  로 놓는다.

- 1.2)  $F_k$ 가 최소인 기계 번호를  $k^*$ 로 놓는다.

Step 2 : 다음과 같은 방법으로 선택된 기계에 배정할 주문을 선택한다.

- 2.1) 기계  $k^*$ 에 배정된 주문이 없다면  $N$ 에 속한 모든 주문  $j$ 에 대해  $s_{jk^*} + p_{jk^*}$ 를 계산하고 그 값을 최소화하는 주문  $j^*$ 를 찾는다.
- 2.2) 그렇지 않으면, 기계  $k^*$ 에 배정된 마지막 주문을  $i$ 로 놓고,  $N$ 에 속한 모든 주문  $j$ 에 대해  $s_{ijk^*} + p_{jk^*}$ 를 계산한다. 그 값을 최소화하는 주문  $j^*$ 를 찾는다.
- 2.3) 주문  $j^*$ 를 기계  $k^*$ 에 배정하고 그 주문을 집합  $N$ 에서 제거한다.
- 2.4)  $M = \emptyset$ 이면 종료하고 그렇지 않으면 Step 1로 간다.

이 알고리즘은 두 부분으로 구성되어 있다. 앞 부분은 기계 군으로부터 하나의 기계를 선택하는 과정이다. 이 과정에서는 ‘가장 먼저 가용해지는 기계를 먼저’ 선택하는 방식을 따른다. 알고리즘의 뒷 부분에서는 선정된 기계에서 처리할 주문을 선택한다. 여기서는 ‘준비시간과 처리시간의 합이 최소인 주문을 먼저’ 배정하는 방식을 적용하였다. 이 알고리즘은 매우 간단하여 쉽게 해를 구할 수 있다. 또한 본 연구에서는 이 알고리즘의 결과를 뒤에서 제시하는 다른 알고리즘과 비교하는 기준으로 사용할 것이다.

그런데 BH 알고리즘은 이해하기 쉽고 해를 구하기도 쉽지만 한번 해가 구해지면 개선이 없는 일방향 알고리즘이므로 그로부터 구해지는 해의 품질을 보증하는 것은 쉽지 않을 것이다. 이러한 인식으로부터 다음에 제시되는 새로운 휴리스틱이 제안되었다. 이 알고리즘의 이름은 IBH(Improved BH) 로 명명하였으며 그 이름에서 알 수 있듯이 이 알고리즘은 BH를 개선한 것이다. 개선방법은 BH에 의해 각 기계에 배정된 주문들을 작업순서가 인접한 주문들끼리 맞바꾸는(pairwise interchange) 것이다. 자세한 절차는 다음과 같다.

##### IBH(Improved Base Heuristic) Algorithm

Step 0 : 초기해는 BH 알고리즘의 결과이다.  $M =$  전체 기계 집합

Step 1 : 집합  $M$ 에서 기계 하나를 임의로 선택하여  $k$ 라고 부른다.

Step 2 : 기계  $k$ 에 연속으로 배정된 주문  $i$ 와  $j$ 에 대해  $\delta_{ij}$  (make-span 감소량)를 다음과 같이 계산한다.

Case 1 : 주문  $i$ 가 기계  $k$ 의 첫 주문이라면,

$$\delta_{ij} = s_{ik} + s_{ijk} - s_{jk} - s_{jik}$$

Case 2 : 주문  $j$ 가 기계  $k$ 의 마지막 주문이라면,

$$\delta_{ij} = s_{ijk} - s_{jik}$$

Case 3 : 둘 다 아니면 주문  $u$ 를  $i$ 의 선행,  $v$ 를  $j$ 의 후속주문이라 할 때  $\delta_{ij} = s_{uik} + s_{ijk} + s_{jvk} - s_{ujk} - s_{jik} - s_{ivk}$

Step 3 : 한 개 이상의  $\delta_{ij}$ 가 양수라면,

- 3.1) 최대의  $\delta_{ij}$ 를 갖는 두 주문의 처리순서를 교환한다.

3.2) Step 2로 간다.

Step 4 : 모든  $\delta_{ij}$  값들이 0이하라면,

4.1) 집합 M에서 기계 k를 제거한다.

4.2) M =  $\emptyset$ 이면 종료하고 그렇지 않으면 Step 1로 간다.

## 5. 유전 알고리즘 방법론

이제 유전 알고리즘 기반의 메타휴리스틱 알고리즘을 개발한다. 전체적인 절차는 일반적인 유전 알고리즘과 대동소이한데, 염색체 표현 및 염색체-해 전환(Decoding) 절차에 있어서 앞 절에서 제안한 휴리스틱의 개념을 도입하였다. 염색체 표현은 주문 수만큼의 (0, 1) 구간 실수를 사용하고, 이 실수 값을 사용해 각각의 주문을 기계에 할당한다. 예를 들어 계획대상 주문이 5개라면 5개의 (0, 1) 구간 실수로 염색체가 구성되고, 첫 번째 실수를 이용해 첫 번째 주문의 기계 할당이, 두 번째 실수를 이용해 두 번째 주문의 기계 할당이, ... 이루어진다. 할당 방법은 각 주문이 모든 기계에 할당되는 비율을 동일하게 유지하기 위하여 구간  $(\frac{k-1}{m}, \frac{k}{m})$ 에 유전자 값이 포함되면 기계 k에 배정하도록 한다. 예를 들어 기계가 3대인 경우 유전자 값이 (0, 1/3)에 속하면 해당 주문을 기계 1에 배정하고, (1/3, 2/3)에 속하면 기계 2에, (2/3, 1)에 속하면 기계 3에 배정하는 것이다. 배정 이후의 일정 생성은 앞선 BH 알고리즘과 같다. n 개의 (0, 1) 구간 실수로부터 하나의 일정이 생성되는 전체 과정을 요약하면 아래와 같다.

### DECODE\_1 Algorithm

Step 1 : 염색체 내부의 유전자 값이 구간  $(\frac{k-1}{m}, \frac{k}{m})$ 에 속하

면 관련되는 주문은 기계 k에 배정한다.

Step 2 : M = 모든 기계들의 집합

Step 3 : M으로부터 기계 하나를 임의로 선택하여 k라고 부른다.

Step 4 : N = 기계 k에 배정된 주문들의 집합

Step 5 : N에 속한 모든 주문 i에 대해  $s_{ik} + p_{ik}$  를 계산하고 그 값을 최소화하는 주문을 기계 k의 첫 주문으로 배정한다. 그 주문을 N에서 제거한다.

Step 6 : 기계 k에 배정된 마지막 주문을 i로 놓고, N에 속한 모든 주문 j에 대해  $s_{ijk} + p_{jk}$  를 계산한 다음 그 값을 최소화하는 주문을 찾아 기계 k의 다음 주문으로 배정하고 그 주문을 N에서 제거한다.

Step 7 : N =  $\emptyset$ 이면 Step 8로 가고 그렇지 않으면 Step 6으로 간다.

Step 8 : 기계 k를 M에서 제거한다.

Step 9 : M =  $\emptyset$ 이면 종료하고 그렇지 않으면 Step 3으로 간다.

이 알고리즘은 앞서 제안한 BH 알고리즘을 염색체-해 전환 과정에 적용한 것이다. 유전자 값에 의해 주문-기계 할당이 이

루어진 다음 BH 알고리즘과 같이 ‘준비시간과 처리시간의 합이 최소인 주문을 먼저’ 처리하는 방식을 적용하였다. 이 방식은 염색체에 의해서는 주문-기계 할당만 이루어지고 한 기계 안에서의 주문처리 순서를 결정하는 것은 휴리스틱에 의해 이루어지므로 일종의 Hybrid 형 유전 알고리즘이라고 할 수 있다.

앞에서 BH를 개선한 IBH 알고리즘을 제안하였듯이 유전 알고리즘 기반의 해법에서도 위의 DECODE\_1 알고리즘을 개선할 수 있다. 개선방법은 IBH 알고리즘과 마찬가지로 각 기계 내에서 주문들의 처리순서를 교환하는 과정을 거치게 된다. 자세한 절차는 아래와 같다.

### DECODE\_2 Algorithm

Step 0 : DECODE\_1을 적용한 해를 갖고 있다. M = 모든 기계들의 집합

Step 1 : M으로부터 기계 하나를 임의로 선택하여 k라고 부른다.

Step 2 : 기계 k에 연속으로 배정된 주문 i와 j에 대해  $\delta_{ij}$  (makespan 감소량)를 다음과 같이 계산한다.

Case 1 : 주문 i가 기계 k의 첫 주문이라면,

$$\delta_{ij} = s_{ik} + s_{ijk} - s_{jk} - s_{jik}$$

Case 2 : 주문 j가 기계 k의 마지막 주문이라면,

$$\delta_{ij} = s_{ijk} - s_{jik}$$

Case 3 : 둘 다 아니면 주문 u 를 i 의 선행, v 를 j 의 후속주문이라 할 때  $\delta_{ij} = s_{uik} + s_{ijk} + s_{jvk} - s_{ujk} - s_{jik} - s_{ivk}$

Step 3 : 한 개 이상의  $\delta_{ij}$ 가 양수라면,

3.1) 최대의  $\delta_{ij}$ 를 갖는 두 주문의 처리순서를 교환한다.

3.2) Step 2로 간다.

Step 4 : 모든  $\delta_{ij}$  값들이 0이하라면,

4.1) 집합 M에서 기계 k를 제거한다.

4.2) M =  $\emptyset$ 이면 종료하고 그렇지 않으면 Step 1로 간다.

위에서 제시한 염색체-해 전환 절차 이외의 과정은 일반적인 유전 알고리즘의 전개와 같다. 대상문제가 최소화 문제이므로 적합도 값( $F_i$  : Fitness Value)은 아래와 같이 모집단 내에서 목적함수 값(즉, 총 완료시간) 중 최대값( $Z_{\max}$ )과 해당 염색체의 목적함수 값( $Z_i$ )의 차이를 사용한다.

$$F_i = Z_{\max} - Z_i \quad (11)$$

다음은 재생(Reproduction) 과정으로 이전 세대의 각 염색체를 그 적합도 값에 따라 다음 세대로 복제하는 절차이다. 본 연구에서는 가장 쉽고도 보편적 방법인 룰렛휠 방법을 사용하였다. 즉, 적합도 값에 비례한 크기의 확률값으로 이전 세대의 염색체를 선택하여 다음 세대의 염색체를 생성하는 것이다. 선택된 염색체를 대상으로 교배(Crossover) 및 돌연변이(Mutation)와 같은 유전 연산(Genetic Operation)을 적용하여 다음 세대의 염색체를 만들어 낸다. 이 유전 연산에는 매우 다양한 방법들

이 있으나 본 연구에서는 가장 쉽고 널리 알려진 방법을 적용한다. 교배는 선택된 두 염색체를 대상으로 한 개의 지점을 선택하여 각각의 유전자를 서로 맞교환하는 방식(one-cut exchange method)을 사용하였다. 또한 돌연변이에서는 임의로 선택된 유전자 값을 (0, 1) 구간의 새로운 난수로 교체하는 방식을 적용하였다. 이러한 재생과정에서는 각 세대의 최우수 염색체가 후속 세대로 넘어가지 않는 경우가 발생할 수 있으므로 본 연구에서는 각 세대의 최우수 염색체 두 개를 선정하여 유전연산을 적용하지 않고 다음 세대로 바로 복사하는 정책(elitist policy)을 사용하였다. 재생과정을 정리하면 아래의 EVOLVE 알고리즘과 같다.

### EVOLVE Algorithm

- Step 1 : 최우수 염색체 두 개는 다음 세대에서 그대로 사용한다.
- Step 2 : 룰렛휠 방법으로 두 개의 염색체를 선택한 다음 (0, 1) 구간의 난수 하나를 생성한다. 이 난수가 미리 정한 교배확률  $P_c$  보다 작으면 위에서 설명한 교배절차에 따라 새로운 염색체 두 개를 생성하고, 그렇지 않으면 두 염색체를 그대로 사용한다. 염색체의 수가 모집단 수에 이를 때까지 이 과정을 반복한다.  $P_c$ 의 값은 실험을 통해 적당한 값을 찾아 사용한다.
- Step 3 : 위의 Step 2에 의해 생성된 염색체에 속하는 모든 유전자에 대해 다음 과정을 수행한다. (0, 1) 구간의 난수를 하나 생성한다. 이 난수가 미리 정한 돌연변이 확률  $P_m$  보다 작으면 (0, 1) 구간에서 새로운 난수를 하나 생성하여 기존의 유전자를 대체한다. 여기서도  $P_m$ 의 값은 실험을 통해 선정한다.

## 6. 수치 실험

앞서 제시한 휴리스틱 알고리즘들의 성능을 평가하기 위해 본 절에서는 임의로 생성한 많은 예제들을 사용해 알고리즘의 결과를 비교한다. 문제의 복잡도가 주문의 수( $n$ )와 기계 수( $m$ )에 가장 크게 영향을 받으므로 이 두 모수들을 몇 개의 수준으로 나누어서 실험을 실시한다. 주문의 수는 20, 30, 40, 50 등의 4개 값 중에서 하나로 하고 기계의 수는 2, 3, 4, 5 등 4개 값 중에서 하나로 하여 총 16개 ( $n, m$ ) 조합의 문제를 생성하여 해법을 적용하였다.

이러한 16개 각각의 ( $n, m$ ) 조합에 대해 10개의 문제를 임의로 생성한다. 문제를 생성하기 위해서는  $a_i, r_k, p_{ik}, s_{ik}, s_{ijk}$  ( $i, j = 1, 2, \dots, n, k = 1, 2, \dots, m$ ) 값들을 생성하여야 한다. 계획대상 주문의 반은 계획시점에 이미 접수되어 있다고 가정하였고 (즉,  $a_i = 0$ ), 나머지 반은  $a_i = U[0, 10]$ 으로 가정하였다. 여기서  $U[a, b]$ 는  $a$ 와  $b$ 사이에서 임의로 생성된 정수라는 의미이다.  $r_k$  값도  $U[0, 10]$ 로 가정한다. 처리시간( $p_{ik}$ )과 준비시간( $s_{ik}$  및  $s_{ijk}$ )은  $U[5, 15]$ 로 가정한다.

생성된 문제들은 앞서 제시한 4개의 알고리즘(BH, IBH 알고리즘 및 DECODE\_1과 DECODE\_2를 적용한 유전 알고리즘)을 사용해 해결한다. 또한 본 연구에서 제시한 알고리즘의 성능을 객관적으로 평가하기 위해 Joo and Kim(2012b)이 제안하였던 알고리즘을 적용하기로 한다. 이 알고리즘도 유전 알고리즘인데 염색체로는 주문들의 처리순서를 표현하고 그 순서에 따라 각 주문을 가장 빠른 시간에 완료될 수 있는 기계를 선택하여 배정하게 된다. Joo and Kim(2012b)은 이 알고리즘을 GA\_DR(Genetic Algorithm with Dispatching Rule)이라고 명명하였으며 매우 뛰어난 성능을 발휘하는 것을 보여주었다.

수치실험의 결과는 <Table 1>에 정리되어 있다. 이 표의 각 값들은 분자를 해당 알고리즘으로 구한 총 완료시간(makespan)으로 하고 분모를 BH 알고리즘으로 구한 총 완료시간으로 하여 계산한 비율값이다. 따라서 값이 작을수록 BH 알고리즘에 비해 작업을 일찍 마쳤다는 의미이므로 알고리즘의 성능이 우수하다고 할 수 있다. 표에서 IBH로 표시된 행은 IBH의 결과와 BH를 비교한 것이고, GA1과 GA2는 각각 DECODE\_1과 DECODE\_2를 적용한 유전 알고리즘 결과를 BH와 비교한 것, 그리고 J&K는 Joo and Kim(2012b)의 GA\_DR과 BH를 비교한 결과이다.

이 표는 16개 ( $n, m$ ) 조합에 따라 16개의 영역으로 이루어져 있다. 각 영역에서는 해당 ( $n, m$ ) 조합에 대해 무작위로 생성된 10개의 문제를 풀 결과를 4개 알고리즘에 대해 정리하였으며 각 알고리즘에 대해서는 3개 종류의 값을 제시하였다. 이 3개 값들은 각 알고리즘으로 10개 문제를 풀고 각 문제에 대해서 위의 비율값을 계산하여 그 평균값, 최소값, 최대값을 찾은 것이다. 예를 들어 표의 좌상귀 셀은 ( $n, m$ ) = (20, 2)인 경우의 IBH 알고리즘에 대한 결과로서 그 값 0.98, 0.92, 1.00은 다음과 같은 절차에 의해 구해졌다.

- 1) 계획대상 작업수( $n$ )는 20이고 기계의 수( $m$ )가 2인 상황에서 생성된 10개의 문제에 대해 BH 알고리즘을 적용한 결과 그 최종 완료시간은 158, 151, 167, 164, 156, 157, 157, 154, 159, 160 등이다.
- 2) 같은 문제들에 대해 IBH 알고리즘을 적용한 최종 완료시간은 146, 151, 160, 164, 153, 157, 157, 151, 159, 156 등이다.
- 3) IBH 결과를 BH 결과로 나누면  $146/158(= 0.92)$ ,  $151/151(= 1.00)$ , ...,  $156/160(= 0.98)$  등이 된다.
- 4) 이 10개의 값들에 대한 평균은 0.98, 최소는 0.92, 최대는 1.00이 된다.

알고리즘들의 성능을 평가하기 위해 표의 값들을 관찰해보면 다음과 같은 내용들을 발견할 수 있다.

- 1) 기계의 수( $m$ )가 적은 경우 본 연구에서 제안한 유전 알고리즘이 Joo and Kim(2012b)의 알고리즘보다 우수하다. 또한 계획대상 작업수( $n$ )가 많아질수록 본 연구에서 제안한 유전 알고리즘이 Joo and Kim(2012b)의 알고리즘보다 우

Table 1. Results of numerical experiments

		n = 20			n = 30			n = 40			n = 50		
		mean	min	max	mean	min	max	mean	min	max	mean	min	max
m = 2	IBH	0.98	0.92	1.00	0.99	0.97	1.00	0.99	0.98	1.01	1.00	0.97	1.01
	GA1	0.93	0.87	0.98	0.95	0.93	1.00	0.95	0.93	0.97	0.98	0.96	1.00
	GA2	0.93	0.87	0.98	0.95	0.92	1.00	0.95	0.92	0.97	0.97	0.95	1.00
	J&K	0.94	0.85	1.01	1.01	0.96	1.05	1.04	1.00	1.08	1.09	1.06	1.12
m = 3	IBH	0.99	0.95	1.03	0.99	0.94	1.01	1.00	0.98	0.01	1.00	0.97	1.01
	GA1	0.94	0.88	0.99	0.95	0.89	1.01	0.99	0.96	1.03	1.01	0.98	1.04
	GA2	0.92	0.87	0.96	0.96	0.89	0.99	0.98	0.95	1.02	1.00	0.97	1.02
	J&K	0.94	0.88	1.01	0.98	0.93	1.01	1.05	1.02	1.08	1.08	1.05	1.13
m = 4	IBH	1.00	0.95	1.07	1.00	0.95	1.05	0.99	0.97	1.01	1.00	0.96	1.04
	GA1	0.94	0.88	0.99	0.99	0.94	1.05	1.03	0.96	1.08	1.05	1.02	1.11
	GA2	0.92	0.86	0.99	0.97	0.93	1.03	1.02	0.94	1.06	1.03	1.00	1.11
	J&K	0.90	0.86	0.97	0.98	0.93	1.05	1.03	0.95	1.09	1.08	1.05	1.14
m = 5	IBH	0.98	0.93	1.00	0.99	0.92	1.03	0.98	0.94	1.00	1.00	0.97	1.02
	GA1	0.91	0.87	0.97	1.01	0.89	1.09	1.04	0.98	1.14	1.08	1.01	1.14
	GA2	0.90	0.86	0.96	0.99	0.92	1.07	1.03	0.97	1.09	1.06	1.01	1.12
	J&K	0.85	0.82	0.90	0.97	0.89	1.04	1.01	0.96	1.07	1.05	1.01	1.10

수해진다. 따라서 기계당 작업수 비율( $n/m$ )이 커질수록 본 연구의 유전 알고리즘이 상대적으로 좋은 성능을 발휘한다고 볼 수 있다.

- 2) 기계의 수( $m$ ) 혹은 계획대상 작업수( $n$ )가 많아질수록 단순 휴리스틱인 IBH의 결과가 유전 알고리즘에 비해 우수해진다. 즉, 문제의 복잡도가 증가하면 유전 알고리즘의 성능이 떨어지게 된다. 물론 모든 경우에 해당되지는 않지만, 단순 휴리스틱이 유전 알고리즘보다 더 우수한 결과를 보여주는 것은 매우 이례적이라 할 수 있다.
- 3) 결론적으로, 기계수( $m$ ) 혹은 작업수( $n$ )가 많은 경우(본 실험에서는  $m \times n$  값이 대략 150 초과)에는 IBH 알고리즘을 사용하고, 그렇지 않은 경우에는 기계당 작업수 비율이 클 때(본 실험에서는  $n/m$  값이 대략 5초과) DECODE\_2를 적용한 유전 알고리즘을, 작을 때는 Joo and Kim(2012b)의 알고리즘을 사용하는 것이 좋다고 할 수 있다.

다음으로는 수행시간의 관점에서 알고리즘의 성능을 살펴본다. 결과는 <Table 2>에 정리되어 있으며 4GB RAM과 2.60GHz G620 CPU가 장착된 PC에서 C++ 언어로 구현된 알고리즘을 가동하였을 때의 소요시간 평균값을 초단위로 표시한 것이다. 이 표도 데이터가 ( $n, m$ ) 조합에 따라 정리되어 있는데, 주의한 점은 단순 휴리스틱(BH 및 IBH)의 수행시간은 표에서 찾아볼 수 없다는 것이다. 단순 휴리스틱의 경우 소요시간이 모두 0.01초 미만이었고 이 값들을 비교하는 것은 현실적으로 큰 의미를 갖기 어렵다고 판단하였기 때문이다.

<Table 2>를 대략적으로 훑어보면 본 연구에서 제시한 유전

알고리즘(GA1 및 GA2)의 소요시간이 대체로 Joo and Kim (2012b)의 GA\_DR보다 길다는 것을 발견할 수 있다. 계획대상 작업의 수( $n$ )가 커질수록 그 차이도 커진다는 것을 알 수 있다. 또한 알고리즘의 종류에 관계없이 계획대상 작업의 수( $n$ )가 커지면 소요시간도 길어지지만 기계의 수( $m$ )는 소요시간에 큰 영향을 미치지 않는다는 것도 이 표를 통해 알 수 있다. 그러나 이러한 소요시간의 변화에도 불구하고 현실적인 관점에서 본다면 단순 휴리스틱은 물론이고 유전 알고리즘 기반의 휴리스틱도 실제상황에서 적용하기에 소요시간이 그리 길지 않다고 할 수 있다.

Table 2. Computation time for GA-based heuristics(seconds)

		n = 20	n = 30	n = 40	n = 50
m = 2	GA1	0.02	0.04	0.10	0.14
	GA2	0.02	0.06	0.11	0.15
	J&K	0.02	0.04	0.06	0.08
m = 3	GA1	0.02	0.06	0.09	0.15
	GA2	0.03	0.07	0.11	0.18
	J&K	0.01	0.01	0.05	0.05
m = 4	GA1	0.04	0.04	0.09	0.10
	GA2	0.03	0.06	0.10	0.14
	J&K	0.01	0.04	0.03	0.09
m = 5	GA1	0.03	0.04	0.09	0.11
	GA2	0.01	0.06	0.08	0.10
	J&K	0.02	0.02	0.04	0.09

## 7. 결 론

본 연구는 플라스틱 사출성형을 전문적으로 처리하는 소규모 제조기업의 사례 문제를 해결하기 위해 처리순서에 따라 작업 준비시간이 달라지는 이종병렬기계 시스템의 일정계획 알고리즘을 개발하였다. 우선 문제의 상황을 잘 표현해주는 수리계획 모형은 기존의 연구와 약간 다른 형태로 제시하였다. 이 최적화 모형은 널리 알려진 것처럼 NP-hard 문제이므로 현실적인 크기의 문제를 해결하기 위해 두 개의 단순 휴리스틱 알고리즘과 두 개의 유전 알고리즘을 개발하였다. 다양한 수치실험을 통해 이 알고리즘들이 비교적 짧은 시간 안에 좋은 품질의 해를 제공해준다는 사실을 확인하였다. 특히 문제에 따라서는 단순 휴리스틱 알고리즘이 유전 알고리즘보다 오히려 나은 해를 제공하기도 하였다.

저자들은 본 연구의 후속으로 다음과 같은 내용을 연구하고 있다. 우선 각 주문의 납기를 고려하는 모형을 개발하고 그 해법을 구한다. 예를 들자면 각 주문의 납기에 대한 지연 혹은 선완료료를 최소화하는 모형이다. 또 하나의 후속연구로는 기계의 운전자에 대한 고려이다. 즉, 한 명의 작업자가 주어진 이종병렬기계 시스템의 작업준비를 모두 담당하는 상황이다. 실제로 자동화가 많이 이루어진 현대의 생산시스템에서 이러한 상황은 쉽게 찾아볼 수 있다.

## 참고문헌

- Agarwal, A., Colak, S., Jacob, V. S., and Pirkul, H. (2006), Heuristics and augmented neural networks for task scheduling with non-identical machines, *European Journal of Operational Research*, **175**(1), 296-317.
- Allahverdi, A., Gupta, J. N. D. and Aldowaisan, T. (1999), A review of scheduling research involving setup considerations, *OMEGA The International Journal of Management Science*, **27**(2), 219-239.
- Allahverdi, A., Ng, C. T., Cheng, T. C. E., and Kovalyov, M. Y. (2008), A survey of scheduling problems with setup times or costs, *European Journal of Operational Research*, **187**(3), 985-1032.
- Cheng, T. C. E. and Sin, C. C. S. (1990), A state-of-the-art review of parallel-machine scheduling research, *European Journal of Operational Research*, **47**(3), 271-292.
- Gharehgozli, A. H., Tavakkoli-Moghaddam, R., and Zaerpour, N. (2009), A fuzzy-mixed integer goal programming model for a parallel-machine scheduling problem with sequence-dependent setup times and release dates, *Robotics and Computer-Integrated Manufacturing*, **25**(4/5), 853-859.
- Guinet, A. (1990), Textile production systems : a succession of non-identical parallel processor shops, *Journal of the Operational Research Society*, **42**(8), 655-671.
- Hop, N. V. and Nagaur, N. N. (2004), The scheduling problem of PCBs for multiple non-identical parallel machines, *European Journal of Operational Research*, **158**(3), 577-594.
- Joo, C. M. and Kim, B. S. (2012a), Non-identical parallel machine scheduling with sequence and machine dependent setup times using meta-heuristic algorithms, *Industrial Engineering and Management Systems*, **11**(1), 114-122.
- Joo, C. M. and Kim, B. S. (2012b), Genetic algorithm with an effective dispatching method for unrelated parallel machine scheduling with sequence dependent and machine dependent setup times, *IE Interfaces*, **25**(3), 357-364.
- Kang, Y. H., Lee, H. C. and Kim, S. S. (1998), Scheduling for non-identical parallel machines with different due dates, *Journal of the Korean Institute of Industrial Engineers*, **24**(1), 37-50.
- Lee, J. S. and Park, S. H. (1999), Scheduling for two stage mixed flow production system with non-identical parallel machines, *Journal of the Korean Institute of Industrial Engineers*, **25**(2), 254-265.
- Li, K. and Yang, S. L. (2009), Non-identical parallel-machine scheduling research with minimizing total weighted completion times : Models, relaxations and algorithms, *Applied Mathematical Modelling*, **33**(4), 2145-2158.
- Marsh, J. D. and Montgomery, D. C. (1973), Optimal procedure for scheduling jobs with sequence-dependent changeover times on parallel processors, *AIIE Technical Papers*, 279-286.
- Potts, C. N. and Kovalyov, M. Y. (2000), Scheduling with batching : A review, *European Journal of Operational Research*, **120**(2), 228-249.
- Ruiz, R. and Vazquez-Rodriguez, J. A. (2010), The hybrid flow shop scheduling problem, *European Journal of Operational Research*, **205**(1), 1-18.
- Tavakkoli-Moghaddam, R., Taheri, F., Bazzazi, F., Izadi, M., and Sassani, F. (2009), Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints, *Computers and Operations Research*, **36**(12), 3224-3230.
- Vallada, E. and Ruiz, R. (2011), A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times, *European Journal of Operational Research*, **211**(3), 612-622.
- Weng, M. X., Lu, J. and Ren, H. (2001), Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective, *International Journal of Production Economics*, **70**(3), 215-226.
- Zhu, Z. and Heady, R. B. (2000), Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach, *Computers and Industrial Engineering*, **38**(2), 297-305.