工業經營學會誌 第 22卷 第 52輯 1999年 11月 109

▩ 연구논문

복수기계를 가진 흐름생산에서 평균흐름시간의 하한에 관한 연구

- A Lower Bound of Mean Flow Time in Shops with Multiple Processors -

김 지 승*
Kim, Ji Seung

Abstract

Flow Shop with Multiple Processors(FSMP) scheduling involves sequencing jobs in a flow shop where, at any processing stage, there exists one or more identical processors. Any methodology to determine the optimal mean flow time for this type of problem is NP-complete. This necessitates the use of sub-optimal heuristic methods to address problems of moderate to large scale. This paper presents global lower bounds on FSMP mean flow time problems which may be used to assess the quality of heuristic solutions when the optimal solution is unknown.

1. 서론

일반적으로 복수기계를 가지는 흐름생산방식(Flow Shop with Multiple Processors)은 n개 작업이 M개 단계를 거쳐 통과하는 흐름생산방식(Flow Shop)의 특성을 나타내며, 각 단계에는 하나 또는 그 이상의 기계가 존재하는 구조이다. 복수기계를 가지는 흐름생산방식(FSMP)문제는 NP-complete로 알려져 있다[5]. 다시 말해서, 복수기계를 가지는 흐름생산방식 문제는 작업의 수(n<9)가 작더라도 최적해를 구하기가 어렵고, 사실상 작업의 수(n>9)가 많으면 최적해를 구하는 것은 거의 불가능하다. 그래서 많은 연구자들은 발견적 접근법으로 최적해는 아니지만 최적해에 가까운 해를 구하는 연구가 이루어지고 있다. 발견적 방법은 비록 최적해를 얻지는 못하지만 적은 계산량으로 빠른 시간내에 근사 최적해를 얻을 수 있다는 장점이 있다.

총 흐름완료시간을 최소화하는 문제를 풀기 위하여 Coffman et al.[3]가 개발한 Multifit로 알려진 발견적 접근법은 bin-packing 개념을 이용하여 복수기계를 가지는 환경에 관한 연구를 하였다. Santos[7] 역시 하나의 단계에서 복수기계(SSMP) 문제를 연구하였다. 2-단계 복수기계를 가진 흐름생산방식에서 2-단계 모두 같은 수의 동일한 기계를 가지는 것을 고려하여 Don E. Deal and John L.Hunsucker[4]는 총 완료시간(Makespan)에 대하여 하한 계산법을 개발하였다.

Narasimhan and Mangiamel[8]는 2-단계 둘 다 동일한 기계를 가진 2-단계 복수기계를 가진 흐름방식(FSMP)에 관하여 연구하였다. Sherali et al.[14]는 2-단계에서 복수기계를 가지는 흐름생산방식에 대하여 다른 평가 기준으로 여러가지 발견적 접근법을 개발하였다.

* 경일대학교 산업시스템공학과

Gupta and Tunc[6]는 LPT에 기초를 둔 발견적 접근법을 개발했고, Johnson은 2-단계 모두 다른 기계를 가지는 2-단계 복수기계를 가진 흐름생산방식에서 총 완료시간을 최소화하는 발견적 접근법을 개발하였다.

Rao[11], Narasimhan and Panwalker[9]는 단계 1에는 하나의 기계를 단계 2에는 두개의 기계를 가지는 2-단계 복수기계를 가지는 흐름생산방식에서 총 완료시간을 최소화하는 발견적 접근법을 개발하였다. 2-단계 이상의 복수기계를 가지는 흐름생산방식에 관한 연구는 Salvador[12]에 의해 처음으로 연구되어졌다. 그는 작업 처리시간을 최소화하는 주기 일정계획 발견적 접근법을 개발하였다. Brah and Hunsucker[1]는 각 단계에 동일한 기계들을 가지는 흐름생산방식에서 총 완료시간을 최소화하는 Branch and Bound 방법을 개발하였다.

Rajendran and Chaudhuri[10]는 복수기계를 가진 흐름생산방식문제에 대해서 총 완료시간을 최소화하는 Branch and Bound 알고리즘을 개발하였다. Brah and Hunsucker and Santos[2]는 다양한 복수기계를 가진 흐름생산방식 구조상에서 총 완료시간과 평균흐름시간을 평가기준으로 하여 9가지의 다른 절차 규칙에 관해 연구를 하였다. 총 완료시간에 대해서는 SPT, MTWF, MWRF규칙이 우수하다고 나타났다. 그러나 평균흐름시간에 대해서는 SPT규칙이 모든 구조에서 우수하게 나타났다. Santos, Hunsucker and Deal[13]는 2-단계 이상의 복수기계를 가지는 흐름생산방식 문제에서 평가기준을 총 완료시간으로 하는 하한을 개발하였다.

이와 같이 대부분의 연구는 단계가 하나 또는 둘인 경우와 평가기준을 총완료시간으로 하는 경우에 대해 이루어졌다. 회사 상황에 따라, 예를 들어 평균적으로 작업의 빠른 회전(rapid turnaround)이 중요한 경우에는 평균흐름시간이 총완료시간보다 더 좋은 성능지표가 된다. 이러한 FSMP의 중요 적용 예로서 하나 이상의 제품을 제조하고 각 작업장에는 여러 대의 기계가 있는 조립라인, 여러 개의 동일한 공장이 있는 석유화학산업, 각 단계에 대체시킬 수 있는 여러 기계가 있는 독립적인 흐름생산 설비들이 있는 공장 등을 들 수 있다. 그러나, FSMP의 평균흐름시간을 최소화하는 연구는 미비하게 연구되었는바, 본 논문에서는 일반적인 FSMP의 총흐름시간에 대한 우수한 하한(Strong Lower Bound)을 개발하는 것이 목적이다. 개발된 하한은 FSMP의 평균흐름시간을 최소화하는 알고리즘의 질을 평가하는데 매우 유용하리라 여겨진다.

2. 총 흐름시간에 대한 하한 유도

복수기계를 가지는 흐름생산방식(FSMP)문제에 대해서, 총 흐름시간의 하한 결정은 각 단계별로 이루어진다. 각 단계별로 생성된 하한 중에서 가장 큰 하한을 문제에 대한 하한으로 결정한다.

j 단계 하한은 LB(j)로 인용되어 진다. 단계 기호는 j로, 범위는 1부터 M까지이다. j 단계에서의 완료시간의 총합은 SUMC(j)로 인용된다. 목적은 SUMC(j)에 대한 하한을 찾는 것이다. 다시 말해서, SUMC(j)를 최소화하는 것이다. 첫 번째 하한은 총 흐름시간에 대한 명백한 하한이다. 다시 말해서, 모든 작업들의 처리시간의 합이며, 이것은 LB(0)로 표기한다. 만약 M 단계를 가진 다중처리기를 가지는 흐름생산방식환경 하에서 진행될 작업이 n개의 있고, 이때 j단계에서, $m(j) \geq 1$ 기계가 (m(j) 는 j단계의 기계수로 단계마다 다를 수 있다) 있다고 하자. 여기서 작업 수 n은 모든 단계에서 기계 m(j)의 수 보다 크다고 가정하며, j단계($1 \leq j \leq M$)에서의 작업 i ($1 \leq i \leq n$)의 처리시간은 $P_{i,j}$ 표기한다. 총 흐름시간은 모든 작업들의 처리시간의 합보다 적을 수가 없다. 그래서 LB(0)는 다음과 같이 계산된다:

$$LB(0) = \sum_{i=1}^{n} \sum_{j=1}^{M} P_{i,j}$$

j 단계 하한치 LB(j)는 다음과 같이 유도한다. 모든 작업들은 처리시간에 관하여 오름차순으

로 순서화되어 진다. 예를 들면, 어떤 단계 j에서 작업 1은 가장 짧은 처리시간을 가진다. 작업 1의 처리시간은 P_{1j}로 인용된다. 단계 j 이전의 총 처리시간은 LS(1,j-1)로 표기하고, 단계 j 이후의 총 처리시간은 RS(1,j+1)로 인용된다. 또한 분할(Partition)의 개념이 하한을 유도하기 위해서 요구된다.

이 분할은 각 단계에서 작업들이 기계에 할당되는 형태를 의미하며 다음과 같이 3가지 경우로 나누어 생각한다.

경우1 : 분할의 수는 만약 n (작업들의 수)이 m(j) (기계들의 수)에 의해 나누어진다면 L (L = n / m(j))이다. 작업순서의 형태는 <그림 1>과 같다.

1 1 ⋯ 1	1 1 ⋯ 1	1 1 ⋯ 1
1	1 :	1 1 ⋯ 1
:	: :	: : ⋯ :
1	1 1	1 1 ⋯ 1
형태 1	형태 k	형태 L

<그림 1> 경우 1의 작업 순서의 형태

경우2 : 분할의 수는 만약 n (작업들의 수)이 m(j) (기계들의 수)에 의해 나누어지지 않고 나머지가 1 이면 L′ (L′ = int(n / m(j)))이다. 작업순서의 형태는 <그림 2>과 같다.

1 1 ⋯ 1	1 1 ⋯ 1	1 1 ⋯ 1
1	1 :	1 ⋯ 1
:	: :	: ⋯ :
1	1 1	1 ⋯ 1
형태 1	형태 k	형태 L′

<그림 2> 경우 2의 작업 순서의 형태

경우3 : 분할의 수는 만약 n (작업들의 수)이 m(j) (기계들의 수)에 의해 나누어지지 않고 나머지가 1보다 크면 L′+1 (L′ = int(n / m(j)))이다. 작업순서의 형태는 <그림 3>과 같다.

1 1 ⋯ 1	1 1 ⋯ 1	1 1 ⋯ 1
1	1 :	: 1 1
:	: :	1 : :
1	1 1	1 1
형태 1	형태 k	형태 L′+1

<그림 3> 경우 3의 작업 순서의 형태

주어진 형태에서, 첫 번째 행은 작업1에서 n_1 까지 첫 번째 기계를 통과하는 작업의 순서를 표현한 것이고, 두 번째 행은 작업 n_1+1 에서 n_1+n_2 까지 두 번째 기계를 통과하는 작업의 순서를 표현한 것이다. 이와 같이 마지막 행은 작업 $n-n_{m(j)}+1$ 에서 n 까지 마지막 기계를 통과하는 기계를 표현한다. <그림1>의 형태에 대해 설명하면, 형태 1은 마지막 기계를 제외한 모든 기계에 하나의 작업만 할당하고 나머지 모든 작업은 마지막 기계에 할당한 형태이다. 형태 L은 모든 기계에 동일한 수의 작업을 할당한 형태이다. (경우1은 작업수가 기계 수에 의해 나누어지므로 가능). 나머지 그림도 같은 개념으로 설명된다.

위의 표기를 근거로 하여, 각 단계별로 하한을 다음 정리에 의해 유도한다.

[정리]

만약 M 단계를 가진 흐름생산방식(FSMP)에서 실행되어져야 할 n개의 작업이 있고, 어떠한 j 단계에서, m(j) (n≥m(j)≥1)개의 독립된 기계들이 있다고 하자. 만약 j단계에서 작업 i의 처리시간이 $P_{i,j}$ 라면, 총 흐름시간의 단계별 하한 LB(j)는 다음과 같다.

LB(j)=minimum(A+B+C)

여기서

$$A=n_1LS(1,j-1)+n_2LS(n_1+1,j-1)+\cdots+n_{m(j)}LS(n-n_{m(j)}+1,j-1)$$

$$B=[n_1P_{1,j}+(n_1-1)P_{2,j}+\cdots+P_{n_1,j}]+[n_2P_{n_1+1,j}+(n_2-1)P_{n_1+2,j}+\cdots+P_{n_1+n_2,j}]$$
$$+\cdots+[n_{m(j)}P_{n-n_{m(j)}+1,j}+(n_{m(j)}-1)P_{n-n_{m(j)}+2,j}+\cdots+P_{n_{m(j)},j}]$$

$$C=\sum_{i=1}^{n_1}RS(i,j+1)+\sum_{i=n_1+1}^{n_1+n_2}RS(i,j+1)+\cdots\cdots+\sum_{i=n-m(j)+1}^{n}RS(i,j+1)$$

(증명)

만약 각 단계에 m 개의 독립된 기계들이 있고, 어떠한 단계에서든, 작업의 수는 기계의 수보다는 크거나 같다고 하면 각 단계마다 n개 작업들을 다음과 같이 분할 할 수 있다. 작업 $n=n_1+n_2+\cdots\cdots+n_m$ 분할은, 작업 1에서 n_1까지는 1기계에서 작업을 하고, n_1+1에서 n_1+n_2 까지는 2기계에서 작업이 이루어지고, 마지막 n_m개의 작업은 마지막 m기계에서 작업이 이루어진다고 하자.

단계 j에서 작업1은 $P_{1,1}+P_{1,2}+\cdots\cdots P_{1,j-1}$ 보다는 일찍 시작할 수 없고, 이것은 LS(1,j-1)로 인용된다. 단계 j상의 작업 1을 끝내기 위해서는 단계 j상의 작업 1의 처리시간인 $P_{1,j}$ 만큼 걸리게 된다. 또한, 작업 1은 $P_{1,j+1}+P_{1,j+2}+\cdots\cdots P_{1,M}$ 보다 더 빨리 나머지 단계를 끝낼 수 없다. 이것은 RS(1, j +1)로 인용된다.

그래서 작업 1의 완료시간은 다음과 같이 표현될 수 있다.

$$C(1,j)\geq LS(1,j-1)+P_{1,j}+RS(1,j+1) \quad (1)$$

같은 방법으로, 작업 2도 다음과 같이 인용된다.

$$C(2,j)\geq LS(1,j-1)+P_{1,j}+P_{2,j}+RS(2,j+1) \quad (2)$$

똑같이 작업 n_1과 작업 n_1+1은 다음과 같다.

$$C(n_1,j)\geq LS(1,j)+P_{1,j}+P_{2,j}+\cdots\cdots P_{n_1,j}+RS(n_1,j+1) \quad (3)$$

$$C(n_1+1,j)\geq LS(n_1+1,j)+P_{n_1+1,j}+RS(n_1+1,j+1)$$

작업 n_1의 흐름시간은 작업 1부터 작업 n_1 까지 각 작업의 완료시간을 합하면 된다. 이 합을 얻기 위해서, 위의 공식들을 합치고 정리하면 다음과 같다.

$$\sum_{i=1}^{n_1}C(i,j)\geq n_1LS(1,j-1)+[n_1P_{1,j}+(n_1-1)P_{2,j}+\cdots\cdots+P_{n_1,j}]+\sum_{i=1}^{n_1}RS(i,j+1)$$

이와 같이, 하나의 분할에 대해서, 이 일정계획에 대한 모든 작업들의 완료시간의 합(이 일정계획에 대한 총 흐름시간)은 다음과 같이 표현된다.

$$TF(j) = \sum_{i=1}^{n} C(i,j) \geq n_1 LS(1, j-1) + [n_1 P_{1,j} + (n_1 - 1)P_{2,j} + \cdots + Pn_{1,j}] + \sum_{i=1}^{n_1} + RS(i, j+1)$$

$$+ n_2 LS(n_1 + 1, j-1) + [n_2 P_{n_1+1,j} + (n_2 - 1)P_{n_1+2,j} + \cdots + P_{n_1+n_2,j}] + \sum_{i=n_1+1}^{n_1+n_2} RS(i, j+1)$$

$$+ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$+ n_{m(j)} LS(n - n_{m(j)} + 1, j-1) + [n_m P_{n-n_{m(j)}+1,j} + (n_m - 1)P_{n-n_{m(j)}+2,j} + \cdots + P_{n,j}] + \sum_{i=n-n_{m(j)}+1}^{n} RS(i, j+1)$$

공식을 간략화하기 위해서, 이 일정계획에 대한 총 흐름시간은 또한 다음과 같이 표현될 수 있다.

SUMC ≥ A + B + C

여기서

$$A = n_1 LS(1, j-1) + n_2 LS(n_1 + 1, j-1) + \cdots + n_{m(j)} LS(n - n_{m(j)} + 1, j-1)$$

$$B = [n_1 P_{1,j} + (n_1 - 1)P_{2,j} + \cdots + P_{n_1,j}] + [n_2 P_{n_1+1,j} + (n_2 - 1)P_{n_1+2,j} + \cdots + P_{n_1+n_2,j}]$$
$$+ \cdots + [n_{m(j)} P_{n-n_{m(j)}+1,j} + (n_{m(j)} - 1)P_{n-n_{m(j)}+2,j} + \cdots + P_{n_{m(j)},j}]$$

$$C = \sum_{i=1}^{n_1} RS(i, j+1) + \sum_{i=n_1+1}^{n_2} RS(i, j+1) + \cdots\cdots + \sum_{i=n_1+n_{m(j)}+1}^{n} RS(i, j+1)$$

각 형태에 대해서 SUMC를 계산한 후, 최소 SUMC가 총 흐름시간에 대한 하한으로 선택된다. 그래서, 단계별 하한 LB(j)는 다음과 같이 구해진다.

LB(j) = minimum (A + B + C) ■

복수기계를 가지는 흐름생산방식(FSMP) 일정계획 문제를 분석하는데 있어서, 총 흐름시간 하한이 단계별로 M+1 (LB(0), LB(1), ……, LB(M)) 개가 구해지고, 이중 최대치가 전체시스템의 하한이 된다. LB(0)은 모든 작업의 처리시간의 합으로 전술한 바와 같다.

흐름생산방식(FSMP) 일정계획 문제에 대한 하한을 LBMAX로 표현하면,

$$LBMAX = \max\{ LB(0), \max_{j} LB(j)\}$$

로 표기할 수 있다.

3. 예제

작업 수:(n)=6 ,각 단계 기계 수:m(1)=m(2)=m(3)=5, 총 단계 수(M)=3 인 경우의 하한을 앞절에서 구한 방법으로 구해보자.

이 경우 분할의 수는 1 (L'=int(n/m(j))=6/5) 인 경우이다.

예제의 처리시간, 단계 수, 기계 수는 <표 1> 에서 보여진다.

<표 1> 각 단계의 작업 처리시

작업 번호	단계 1 (기계 수=5)	단계 2 (기계 수=5)	단계 3 (기계 수=5)
1	4	8	5
2	2	1	3
3	3	5	4
4	6	6	9
5	5	8	7
6	3	5	7
합	23	33	35

LB(0)=23+33+35=91 이다.
각 단계에 대해서 오름차순으로 순서화한다.
처리시간의 새로운 순서는 <표 2>와 같다.

<표 2> 각 단계별 작업의 처리시간을 오름차순으로 순서화

작업순서 1	2	1	3	
작업순서 2	3	5	4	
작업순서 3	3	5	5	
작업순서 4	4	6	7	
작업순서 5	5	8	7	
작업순서 6	6	8	9	
합	23	33	35	91

단계 1,2,3에서 분할의 형태가 각 한 가지 형태로 나타난다.

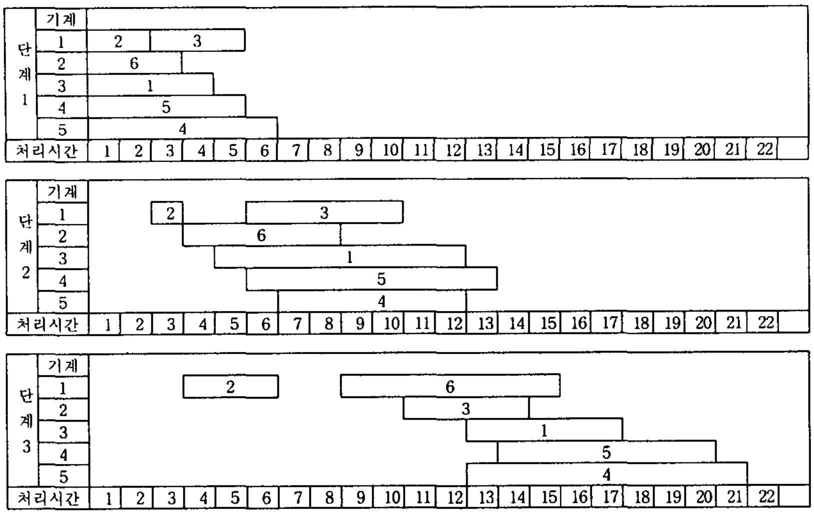
단계 1	단계 2	단계 3
② ③	① ⑤	③ ④
③	⑤	⑤
④	⑥	⑦
⑤	⑧	⑦
⑥	⑧	⑨
형 태 1	형 태 1	형 태 1

단계 1,2,3을 계산하면

단계 1 계산	단계 2 계산	단계 3 계산
A=0	A=2*2+1*(3+3+4+5) =4+15=19	A=19+2*1+1*(5+5+6+8) =19+2+24=45
B=2*2+1*(3+3+4+5+6) =4+21=25	B=2*1+1*(5+5+6+8+8) =2+32=34	B=2*3+1*(4+5+7+7+9) =6+32=38
C=33+35 =68	C=35	C=0
SUMC=0+25+68 =93	SUMC=19+34+35 =88	SUMC=45+38+0 =83
LB(1) =93	LB(2)=88	LB(3)=83

∴LBMAX=max{ 91,93,88,83}=93

<표 3>예제 문제의 최적 일정계획



각 작업의 흐름시간은 다음과 같다.

작업1=17, 작업2=6, 작업3=14, 작업4=21, 작업5=20, 작업6=15이다.

그러므로 총흐름시간은 각 작업의 흐름시간 합으로 93이 된다. 위에서 하한으로 구한 값과 동일함을 보여주고 있다. 즉 본 예제에서의 하한값은 최적치임을 알 수 있다.

4. 결론

본 논문에서는 복수기계를 가진 흐름생산방식(FSMP)에서 총흐름시간(Total Flow time)의 하한을 개발하였다. FSMP 환경하에서 평균흐름시간을 최소화하는 일정계획을 구하는 것은 NP-complete로서 최적 일정계획을 구하는 것은 거의 불가능하다. 따라서 발견적 기법을 연구하는데 초점이 맞추어져 있다. 이 경우 최적해와 비교를 하기가 전술한 바와 같이 쉽지 않다. 본 연구에서 구한 하한이 이 경우 발견적 기법의 해를 평가하는데 매우 유용하리라 여겨진다. 예제에서는 본 연구의 하한과 최적값이 같이 나왔다. 현재까지의 실험결과는 약 67%가 최적값과 하한이 같은 값을 가졌다. 또한 나머지도 최적치와 10% 미만인 것으로 나타났다. 따라서 본 논문의 방법으로 구한 하한은 최적값에 매우 근접하여 좋은 평가지표가 되리라 생각된다. 실험대상으로는 단계를 3에서 5, 각 단계별 기계수를 2에서 5, 작업수를 5에서 10으로 변환시키며 일반적인 FSMP의 500문제를 풀어보았다.

앞으로의 과제는 FSMP 환경하에서 평균흐름시간을 최소화하는 효율적인 발견적 기법을 개발하고, 또한 개발한 방법으로 전술된 하한을 좀더 최적치에 가깝게 보완하는 작업으로서 현재 연구중이다.

참 고 문 헌

[1] Brah, S.A. and Hunsucker, J.L. 1991. Branch and Bound Algorithm for the Flow Shop with Multiple Processors. European Journal of Operation Research, Vol. 51, pp88-99.

[2] Brah, S.A., Santos, D.L., and Hunsucker, J.L. 1989. Heuristic Programming Study of a Flow Shop with Multiple Processors scheduling. Paper Presented at the ORSA/TIMS Joint National Meeting in New York City.

[3] Coffman, E.G. Jr., Garey , M.R., and Johnson, D.S. 1978. An Application of Bin Packing to Multiprocessor Scheduling. SIAM Journal on Computing, Vol.7, 1-17.
[4] Deal, D.E., and Hunsucker, John.l. 1991. The Two-Stage Flowshop Scheduling Problem with M Machines at each Stage. Journal of Information & Optimization Sciences, Vol 12 (1991), No.3, pp407-417.
[5] Garey, M.R., and Johnson, D.S. 1979. Computers and Intractability: A Guide to the Theory of NP-Comleteness, Freeman Press, San Francisco.
[6] Gupta, J.N.D., and Tunc, E.A. 1991. Schedules for a Two-Stage Hybrid Flowshop with Parallel Machines at the Second Stage. International Journal of Production Research. Vol. 29, pp1489-1502.
[7] Johnson, S., "Optimal Two- and Three-stage production schedules with set up times included", Naval Research Logistics Quarterly, Vol. 1 (1954), pp.61-68.
[8] Narsimhan, S.L., and Mangiameli, P.M. 1987. A Comparison of Sequencing Rules for a Two-Stage Hybrid Flow Shop. Decision Sciences. Vol. 18, pp250-265.
[9] Narsimhan, S.L., and Panwalker, S.S. 1984. Sequencing in a Two-Stage Manufacturing Process. International Journal of Production Research. Vol. 22, pp555-564.
[10] Rajendran, c. and Chaudhuri, d. 1992. A Multi-Stage Parallel-Processor Flowshop Problem with Minimum Flowtime. European Journal of Operation Research, Vol. 57, pp111-122.
[11] Rao, T.B.K. 1989. Sequencing in the Order A, B with Multiplicity of Machines for a single Operation. Omega. Vol. 17 pp551-557.
[12] Salvadar, M.S. 1973. "A Solution to Special Case of Flow Shop Sequencing Problem", in: S.E. Elmaghraby (ed.), Symposium of the Theory of Sequencing and Applications, Springer-Verlag, New York.
[13] Santos, D.L., and Hunsucker, J.L. Deal, D.E. 1995a. Global Lower Bounds for Flow Shops with Multiple Processors. European Journal of Operation Research, Vol. 80, pp112-120.
[14] Sherali, H.D., Sarin, S.C., and Kodiacam, M.S. 1990. Models and Algorithms for a Two-Stage Production Process. Production Planning and Control, Vol. 1, pp27-39.