經營科學
第18卷 第1號
2001年 5月

단일기계에서 총납기지연 최소화를 위한 작업쌍 비교 규칙의 이행성 증명

전태준* · 박성호**

A Proof of Transitivity of Job-Pair Comparison Rule to Minimize Total Tardiness in a Single Machine

Tae-Joon Jeon* · Sung-Ho Park**

■ Abstract ■

In this paper, We propose the Job-Pair Comparison (JPC) rule to minimize total tardiness in a single machine. For this purpose we derive conditional expression to choose the desirable job sequence. We also prove the transitivity of JPC rule, to prevent cycle and to decrease the complexity of comparison.

1. 서 론

동일 시작시간을 갖는 단일기계(Single Machine) 문제에서 총납기지연(Total Tardiness) 최소화를 위하여 Baker[2]는 두 개 Job 간의 서로 상반된 순서관계에 대한 총납기지연 정도를 비교하는 방법을 사용하였다. 이 과정에서 Baker[2]는 어느 한가지 할당규칙만으로는 총납기지연을 최소화하는 작업(Job)의 선택이 불가능하다는 사실을 보였고, 이에 따라 비교되는 두 작업이 갖는 가공시간과 납기 변수간의 관계에 따라 구분되는 전체 구조를 정리하여 조건에 따라 EDD(Earliest Due Date)와 SPT(Shortest Processing Time) 할당규칙을 달리 적용되어 선행이 바람직한 작업이 선택됨을 보였다. 그리고 이후에 이 조건식을 이용하는 기법에 대한 많은 연구들이 진행되어져 왔다[1, 3-7].

전체 스케쥴 대상 작업들을 두 개 작업씩 비교하여 Sequence를 구해 나가는 과정에서, 사용되는

* 전남대학교 산업공학과
** 전남도립 담양대학 초고속정보통신공학부

총납기지연 값은 작업의 단일 특성만을 고려되는 기존의 일반적 할당규칙과 달리 계산과정에 두 개 작업간의 다수 특성이 함께 반영되어 나타난 정보이기 때문에 상대적으로 좋은해를 구할 수 있다.

그러나 작업을 쌍으로 비교해 나가는 방법은 최종 Sequence를 구하기까지 상대적으로 많은 비교 횟수가 요구되는 문제점과 비교되는 작업들 중에서 선행 작업을 결정하지 못하고 순환이 발생될 수 있는 문제점이 있다. 이런 문제점을 위해 비교되는 다수 작업 들간의 이행성(Transitivity) 성립에 대한 증명이 요구된다.

따라서 본 논문에서는 작업쌍 비교(Job Pair Comparison) 할당규칙을 제시하고, 이 할당규칙을 적용하여 선택된 작업들 간에 이행성(Transitivity)이 성립됨을 증명함으로서 작업 선택과정에서의 순환방지 및 비교를 감소시키고자 하였다.

본 논문의 구성은 먼저, 2장에서 작업쌍 비교 할당규칙을 설명하고, 3장에서 이에 대한 이행성(Transitivity) 증명과정을 보였다. 그리고 결론 및 추후 연구는 4장에서 언급 하였다.

2. 작업쌍 비교(Job Pair Comparison) 할당규칙

작업쌍 비교 할당규칙은 스케쥴 대상 작업들 중에서 전체 Sequence 상에서 가장 선행되야 하는 작업을 결정하기 위하여, 작업들을 두 개 작업씩 쌍으로 묶어 상반된 Sequence를 비교해 나가는 방법을 사용한다. 즉, 더 작은 총납기지연이 발생되는 Sequence에서의 선행 작업이 선택된 뒤, 이 작업이 또 다른 모든 작업과의 비교과정을 통하여 최종 선행되야 하는 작업을 결정한다.

작업쌍 비교 할당규칙의 설명을 위하여 먼저, 두 개의 작업 i와 j가 서로 다른 순서관계로 비교될 때, 납기지연 형태에 따라 발생 가능한 모든 Case를 구분해보면 다음 <표 1>과 같이 정리된다.

〈표 1〉 작업쌍 비교 Case

Case		Sequence $i \rightarrow j$		Sequence $j \rightarrow i$	
1	1.1	Early	Early	Early	Early
	1.2	Early	Early	Early	Tardy
	1.3	Early	Tardy	Early	Early
	1.4	Early	Tardy	Early	Tardy
2	2.1	Early	Tardy	Tardy	Early
	2.2	Early	Tardy	Tardy	Tardy
3	3.1	Tardy	Early	Early	Tardy
	3.2	Tardy	Tardy	Early	Tardy
4		Tardy	Tardy	Tardy	Tardy

그러면 각 Case 별로 선행 작업이 결정되는 과정을 살펴보면 다음과 같다. 여기서 t는 작업 i와 j의 시작가능 시점을 나타낸다.

Case 1. $t+p_i < d_i$ and $t+p_j < d_j$

Case 1.1 $t+p_i+p_j < d_j$ and $t+p_j+p_i < d_i$

모두 납기지연 되지 않는 경우로서 임의의 작업 선택이 가능하지만, 본 연구에서는 두 작업의 납기관계가 $d_i < d_j$일 경우는 작업 i가 선행되고, 반대로 $d_i \geq d_j$일 경우는 작업 j가 선행되는 것으로 본다.

Case 1.2 $d_i \leq t+p_i+p_j < d_j$

작업 i가 후행되면 납기지연이 발생되므로 항상 작업 i가 선행되어야 하는 경우이고 이 때 $d_i < d_j$ 관계는 항상 성립된다.

Case 1.3 $d_j \leq t+p_i+p_j < d_i$

Case 1.2와는 반대로 작업 j가 후행될 때 납기지연이 발생되는 경우이므로 항상 작업 j가 선행되어야 하며 이 때 $d_i > d_j$ 관계가 항상 성립된다.

Case 1.4 $t+p_i+p_j \geq d_j$ and $t+p_j+p_i \geq d_i$

작업 i와 j가 후행될 때 모두 납기지연이 발생되므로, 두 순서관계에서의 총납기지연 값을 비교하면

$$T_{ij}-T_{ji}=(t+p_i+p_j-d_j)-(t+p_j+p_i-d_i) = d_i-d_j$$

이다.

따라서 만약 $d_i < d_j$이면 $T_{ij} < T_{ji}$에 의해서 작업 i가 선행되어야 하고, 그렇지 않고 $d_i \geq d_j$ 이면 작업 j가 선행되어야 한다.

결국 Case 1.에서는 작업 i와 j의 후행위치에서의 납기지연 여부에 따라 4가지의 세분된 경우로 구분해서 살펴 본 결과 $d_i < d_j$일 경우는 작업 i가, $d_i \geq d_j$일 경우는 작업 j가 선행되어야 한다.

Case 2. $t+p_i < d_i$ and $t+p_j \geq d_j$

먼저, $t+p_j \geq d_j$에 의해 $t+p_i+p_j \geq d_j$ 관계가 설명되어진다. 그러므로 작업 i가 후행될 때의 납기지연 여부에 따라 구분해보면,

Case 2.1 $t+p_j+p_i < d_i$

여기서는 다음과 같은 이유로 작업 j가 항상 선행되는 경우이다.

$$T_{ij}-T_{ji}=(t+p_i+p_j-d_j)-(t+p_j-d_j) = p_i \geq 0$$

그리고 $t+p_j+p_i < d_i$ 식과 $t+p_j < t+p_j+p_i$ 식에 의해서 $d_i > t+p_j$ 관계가 항상 성립된다.

Case 2.2 $t+p_j+p_i \geq d_i$

두 순서관계 에서의 납기지연 값을 비교해보면,

$$T_{ij}-T_{ji}=(t+p_i+p_j-d_j)-\{(t+p_j-d_j)-(t+p_j+p_i-d_i)\} = -(t+p_j-d_i)$$

이다. 따라서 $d_i < t+p_j$ 관계일 경우는 $T_{ij} < T_{ji}$가 되므로 작업 i가 선행되어야 하고 반대로 $d_i \geq t+p_j$ 관계일 경우는 $T_{ij} \geq T_{ji}$가 되므로 작업 j가 선행된다.

결국 Case 2.는 $d_i < t+p_j$일 경우는 작업 i가 선행되고, $d_i \geq t+p_j$일 경우는 작업 j가 선행된다.

Case 3. $t+p_i \geq d_i$ and $t+p_j < d_j$

먼저, $d_i < t+p_i$에 의해서 $d_i < t+p_i+p_j$ 관계는 항상 성립된다. 그러므로 작업 j가 후행될 때의 납기지연 여부에 따라 구분해보면,

Case 3.1 $t+p_i+p_j < d_j$

여기서는 항상 작업 i가 선행되는 경우로서 먼저, $p_i < p_j$이면 $t+p_i < t+p_j < d_j$ 관계에 의해서 $t+p_i < d_j$가 항상 성립된다. 또 다른 $p_i \geq p_j$ 관계에서도 $t+p_i < t+p_i+p_j < d_j$에 의해서 마찬가지로 $t+p_i < d_j$가 항상 성립되고 있다.

Case 3.2 $d_j < t+p_i+p_j$

두 순서관계에서의 총납기지연 값을 비교해보면,

$$T_{ij}-T_{ji}=(t+p_i-d_i)+(t+p_i+p_j-d_j)-(t+p_j+p_i-d_i) = t+p_i-d_j$$

이므로 만약, $t+p_i < d_j$이면 $T_{ij} < T_{ji}$에 의해서 작업 i가 선행되고, 그렇지 않고 $t+p_i \geq d_j$이면 작업 j가 선행되야 한다.

결국 Case 3.에서는 $t+p_i < d_j$ 이면 작업 i가 선행되야 하고, $t+p_i \geq d_j$ 이면 작업 j가 선행되야 한다.

Case 4. $t+p_i \geq d_i$ and $t+p_j \geq d_j$

여기서는 모든 순서관계에서 납기지연이 발생되는 경우로서 총납기지연 값을 비교해보면,

$$\begin{aligned} T_{ij}-T_{ji} &= (t+p_i-d_i)+(t+p_i+p_j-d_j) \\ &\quad -(t+p_j-d_j)-(t+p_j+p_i-d_i) \\ &= p_i-p_j \end{aligned}$$

이다. 따라서 $p_i < p_j$ 일 경우, 즉 $t+p_i < t+p_j$ 이면 $T_{ij} < T_{ji}$ 에 의해서 작업 i가 선행되고, 그렇지 않고 $t+p_i \geq t+p_j$ 이면 작업 j가 선행되야 한다.

결국 모든 Case에 대하여 선행되는 작업을 결정하여 정리하면 <표 2>과 같이 전체구조가 간략화 된다. 여기서 편의상 작업 i와 j의 비교 시 작업 i 선행되는 것이 바람직한 경우를 표현하기 위하여 $(i \rightarrow j)$ 기호를 사용하기로 하고, 그 반대의 경우는 $(j \rightarrow i)$ 기호를 사용하기로 한다.

<표 2>의 정리된 결과를 보면 작업 i와 j의 선행위치에서 납기지연 여부에 따라 구분된 4가지 Case들은 앞에서 세분하여 살펴 본 Case별로 공통되게 선행되는 작업을 결정하는 조건식이 존재한다. 그런데 이 조건식들은 각 작업의 선행위치에서의 납기지연 여부를 나타내고 있다. 예를 들어서 <표 2>의 첫 번째 경우를 보면, 작업 i의 선행위치에서 납기지연 여부를 나타내는 $t+p_i < d_i$ 조건식과 작업 j의 선행위치에서 납기지연 여부를 나타내는 $t+p_j < d_j$ 조건식에서 최종적으로 선행되는 작업을 결정하는 조건식은 d_i 와 d_j 의 크기관계이다. 그런데 이것은 첫 번째 경우의 각 작업의 납기지연 여부를 나타내는 조건식에서 더 큰 값들 간의 크기 관계를 비교하는 것과 같다.

<표 2>의 다른 3가지 경우에서도 마찬가지로 선행위치에서의 납기지연 여부를 나타내고 있는 조건식들에서 더 큰 값을 갖는 조건식간의 크기 관계를 비교하는 것이 곧 최종 선행 작업을 결정하는 조건식의 관계와 같게 나타나고 있다.

결국 <표 2>의 정리된 결과는 다음 식과 같이 두 가지 경우로 표현 가능하다.

(1) $\max\{d_i,\ t+p_i\} < \max\{d_j,\ t+p_j\}$ 이면 $(i \rightarrow j)$

(2) $\max\{d_i,\ t+p_i\} \geq \max\{d_j,\ t+p_j\}$ 이면 $(j \rightarrow i)$

〈표 2〉 Case 별 순서관계

Case	조 건		순 서
1	$t+p_i < d_i,\ \ t+p_j < d_j$	$d_i < d_j$	$(i \rightarrow j)$
		$d_i \geq d_j$	$(j \rightarrow i)$
2	$t+p_i < d_i,\ \ t+p_j \geq d_j$	$d_i < t+p_j$	$(i \rightarrow j)$
		$d_i \geq t+p_j$	$(j \rightarrow i)$
3	$t+p_i \geq d_i,\ \ t+p_j < d_j$	$t+p_i < d_j$	$(i \rightarrow j)$
		$t+p_i \geq d_j$	$(j \rightarrow i)$
4	$t+p_i \geq d_i,\ \ t+p_j \geq d_j$	$t+p_i < t+p_j$	$(i \rightarrow j)$
		$t+p_i \geq t+p_j$	$(j \rightarrow i)$

따라서 위 조건식은 각 작업에서의 선행위치에서의 납기지연 여부와 그에 따른 최종 선행 작업의 선택을 결정하는 조건식이 반영되어 나타난 정보라고 볼 수 있고 하나의 할당규칙으로 생각할 수 있다. 이를 본 논문에서는 작업쌍 비교 할당규칙(Job Pair Comparison : 이후 JPC) 이라고 명명하였다.

3. Transitivity 증명

앞에서도 언급했듯이 JPC 할당규칙은 적용과정에서 이행성(Transitivity)이 만족됨을 증명해야 된다. 만약 그렇지 못하다면, 작업쌍 비교 방법은 첫째, 비교과정에서 탈락된 작업들에 대하여 비교 과정에서 제외시키지 못함으로 인하여 ${}_nC_2$의 많은 비교횟수가 요구될 수 있기 때문에 할당규칙이 갖는 단순성이 결여될 것이고 둘째, 스케쥴 대상 작업 간의 바람직한 순서관계를 결정할 수 없는 순환 상황의 발생 가능성도 생각할 수 있다.

따라서 여기서는 JPC 할당규칙의 적용과정에서 이행성(Transitivity)이 성립됨을 증명하고자 한다.

그러면 이를 위하여 임의로 세 개의 작업 i, j 그리고 k를 고려하고, 다음과 같이 정의하자.

(정의. 1)

세 개의 작업 i, j 그리고 k에 대하여 JPC 할당규칙을 적용했을 때, $(i \to j)$이고 $(j \to k)$인 경우 $(i \to k)$의 관계가 성립된다면 이행성(Transitivity)이 성립된다.

증명을 위하여 <표 2>에서 $(i \to j)$가 되는 조건들만을 고려하면 아래와 같이 4가지 Case가 존재하고, 마찬가지로 $(j \to k)$에 대해서도 4가지 Case가 존재함을 알 수 있기 때문에 결국 총 16가지 Case에 대한 증명이 요구된다.

Condition $(i \to j)_{①}$:

$$t+p_i < d_i\ ,\ t+p_j < d_j \text{ and } d_i < d_j$$

Condition $(i \to j)_{②}$:

$$t+p_i < d_i\ ,\ t+p_j \geq d_j \text{ and } d_i < t+p_j$$

Condition $(i \to j)_{③}$:

$$t+p_i \geq d_i,\ t+p_j < d_j \text{ and } t+p_i < d_j$$

Condition $(i \to j)_{④}$:

$$t+p_i \geq d_i,\ t+p_j \geq d_j \text{ and } t+p_i < t+p_j$$

그런데 16가지 Case에서도 비교되는 과정에서 작업 j가 갖는 납기지연 형태가 서로 모순되는 8가지는 발생되지 않는 Case로서 증명에서 제외된다.

그리고 각 Case에 대한 Transitivity 증명을 수행하면 다음과 같다.

3.1 Condition $(i \to j)_{①}$ & Condition $(j \to k)_{①}$

먼저, 선행위치에서 작업 i와 k가 납기지연이 발생되지 않는 $t+p_i < d_i$와 $t+p_k < d_k$인 관계를 알 수 있다. 그러므로 후행위치 에서 납기지연 되는 여부에 따라 구분한다.

3.1.1 $t+p_i+p_k < d_k$ and $t+p_k+p_i < d_i$

모두 납기지연이 발생되지 않는 경우로서 같은 납기지연을 갖지만, 결합관계에서 유도되는 조건식 $d_i < d_k$에 의해서 Condition $(i \to k)_{①}$으로 되므로, 작업 i가 선행되는 것이 더 좋다.

3.1.2 $d_i < t+p_i+p_k < d_k$

항상 작업 i가 k에 선행되는 경우로서 후행위치의 납기지연 여부를 나타내는 위의 조건식으로부터의 $d_i < d_k$ 관계가 항상 만족되므로 이것은 결합관계에서 유도된 조건식과 일치된다. 따라서 Condition $(i \to k)_{①}$으로 되므로, 작업 i가 선행된다.

3.1.3 $d_k < t+p_i+p_k < d_i$

항상 작업 k가 i에 선행되는 경우이고 후행위치의 납기지연 여부를 나타내는 위의 조건식으로부

터의 $d_i \geq d_k$ 관계가 항상 만족되고 있다. 하지만 이 경우는 결합관계에서 유도되는 조건식 $d_i < d_k$ 관계와 모순되며 발생될 수 없는 경우이다.

3.1.4 $t+p_i+p_k \geq d_k$ and $t+p_k+p_i \geq d_i$

작업 i와 k가 후행될 때 모두 납기지연이 발생되므로 서로의 총납기지연 값을 비교해보면

$$T_{ik} - T_{ki} = (t+p_i+p_k-d_k)-(t+p_i+p_k-d_i) = d_i - d_k$$

가 된다. 이 경우는 결합관계에서 유도되는 조건식 $d_i < d_k$ 관계에 의해 Condition $(i \rightarrow k)_{①}$으로 되므로, 작업 i가 선행된다.

3.2 Condition $(i \rightarrow j)_{①}$ & Condition $(j \rightarrow k)_{②}$

먼저, 선행위치에서 작업 i와 k의 납기지연여부를 보면 $t+p_i < d_i$와 $t+p_k \geq d_k$인 관계를 알 수 있고 더불어 $t+p_i+p_k \geq d_k$ 관계 역시 알 수 있다. 그러므로 후행위치에서 작업 i의 납기지연 여부에 따라 구분한다.

3.2.1 $t+p_k+p_i < d_i$

작업 k가 작업 i에 항상 선행되는 경우이다.

$$T_{ik} - T_{ki} = (t+p_i+p_k-d_k)-(t+p_k-d_k) = p_i \geq 0$$

그러나 이 경우는 작업 k가 선행되고 i가 후행될 때의 납기지연을 나타내는 두 식을 한 식으로 써보면 $d_k < t+p_k < t+p_k+p_i < d_i$, 즉 $t+p_k < d_i$와 같은 관계가 되는데 이 경우는 결합관계에서 유도되는 조건식 $d_i < t+p_k$에 모순되는 조건식으로 발생될 수 없는 경우이다.

3.2.2 $t+p_k+p_i \geq d_i$

서로 다른 순서관계에서의 총납기지연 값을 비교해보면,

$$\begin{aligned} T_{ik} - T_{ki} &= (t+p_i+p_k-d_k) \\ &\quad - \{(t+p_k-d_k)+(t+p_k+p_i-d_i)\} \\ &= -(t+p_k-d_i) \end{aligned}$$

이다. 이 경우는 결합관계에서 유도되는 조건식 $d_i < t+p_k$에 의하여 Condition $(i \rightarrow k)_{②}$로 되므로, 작업 i가 선행된다.

3.3 Condition $(i \rightarrow j)_{③}$ & Condition $(j \rightarrow k)_{①}$

먼저, 선행위치에서 작업 i와 k가 $t+p_i \geq d_i$와 $t+p_k < d_k$인 관계를 알 수 있고, $t+p_k+p_i \geq d_i$ 관계 역시 설명된다. 그러므로 작업 k가 후행위치에서 납기지연 여부에 따라 구분해보면,

3.3.1 $t+p_i+p_k < d_k$

항상 작업 i가 작업 k에 선행되는 경우이다.

$$T_{ik} - T_{ki} = (t+p_i-d_i)-(t+p_k+p_i-d_i) = -p_k < 0$$

그리고, 이 경우는 작업 i가 선행되고 작업 k가 후행될 때의 납기지연을 나타내는 두 식을 한 식으로 써보면 $d_i < t+p_i < t+p_i+p_k < d_k$, 즉 $t+p_i < d_k$와 같은 관계가 되며 이것은 결합관계에서 유도되는 조건식 $t+p_i < d_k$와 일치된다. 따라서 Condition $(i \rightarrow k)_{③}$으로 되므로, 작업 i가 선행된다.

3.3.2 $t+p_i+p_k \geq d_k$

서로 다른 순서관계에서의 총납기지연 값을 비교해보면,

$$T_{ik} - T_{ki} = \{(t+p_i-d_i)+(t+p_i+p_k-d_k)\} - (t+p_k+p_i-d_i) = (t+p_i-d_k)$$

이다. 이것은 결합관계에서 유도되는 조건식 $t+p_i < d_k$에 의하여 Condition $(i \rightarrow k)_{③}$으로 되므로, 작업 i가 선행된다.

3.4 Condition $(i \rightarrow j)_{③}$ & Condition $(j \rightarrow k)_{②}$

모든 위치에서 작업들이 납기지연 되는 경우로서 두 순서관계에서의 납기지연 값을 비교하면,

$$T_{ik} - T_{ki} = \{(t+p_i-d_i)+(t+p_i+p_k-d_k)\} - \{(t+p_k-d_k)+(t+p_k+p_i-d_i)\} = p_i - p_k$$

이다. 이것은 결합관계에서 유도되는 조건식 $t+p_i < t+p_k$, 즉 $p_i < p_k$에 의해서 Condition $(i \rightarrow k)_{④}$로 되므로, 작업 i가 선행된다.

위와 증명과정이 나머지 4가지 결합 Case에서도 동일하게 적용되어 결국 <표 3>과 같이 모든 경우에서 이행성(Transitivity)이 성립됨을 증명하였다.

〈표 3〉 Transitivity 증명

$(i \rightarrow j)$ \ $(j \rightarrow k)$	$(j \rightarrow k)_{①}$	$(j \rightarrow k)_{②}$	$(j \rightarrow k)_{③}$	$(j \rightarrow k)_{④}$
$(i \rightarrow j)_{①}$	$(j \rightarrow k)_{①}$	$(j \rightarrow k)_{②}$	×	×
$(i \rightarrow j)_{②}$	×	×	$(j \rightarrow k)_{①}$	$(j \rightarrow k)_{②}$
$(i \rightarrow j)_{③}$	$(j \rightarrow k)_{③}$	$(j \rightarrow k)_{④}$	×	×
$(i \rightarrow j)_{④}$	×	×	$(j \rightarrow k)_{③}$	$(j \rightarrow k)_{④}$

4. 결 론

본 논문에서는 동일 시작시간을 갖는 Single Machine 문제에서 총납기지연을 최소화하기 위하여 두 개 작업씩 쌍으로 비교하는 새로운 JPC 할당규칙을 제시하였고, 이 규칙이 적용되는 과정에서 비교 작업 간의 이행성(Transitivity)이 만족됨을 보임으로서 순환이 발생되지 않음을 보였다.

현재 이 개념을 다른 시작시간을 갖는 Single Machine 문제에 대해 적용하기 위한 연구가 진행되고 있으며, 더불어 작업 Shop 문제를 대상으로 하는 확장 연구가 함께 진행되고 있다.

참 고 문 헌

[1] Alidaee, B. and Gopalan, S., "A Note on the Equivalence of Two Heuristics to Minimize Total Tardiness," EJOR 96, pp.514-517, 1997.

[2] Baker, K.R., "Introduction to Sequencing and Scheduling," John Wiley & Sons, 1974.

[3] Baker, K.R. and J.W. Bertrand, "A Dynamic Priority Rule for Scheduling Against Due-dates," Journal of Operations Management 3, pp.37-42, 1982.

[4] Holsenback, J.E., "A Pseudopolynomial Algorithm for Sequencing on One Machine to Minimize Total Tardiness," Journal of the Operational Research Society 43, pp.53-62, 1992.

[5] Panwalkar, S.S., M.L. Simth and C.P. Koulamas, "A Heuristic for the Single Machine Tardiness Problem," EJOR 70, pp.304-310, 1993.

[6] Russell, R.M. and J.E. Holsenback, "Evaluation of Leading Heuristic for the Single Machine Tardiness Problem," EJOR 96, pp.538-545, 1997.

[7] Wilkerson, J. and J.D. Irwin, "An Improved Algorithm for Scheduling Independent Tasks," AIIE Transactions 3, pp.333-342, 1971.