An Improvement of the Lower Bound in Flowshop Scheduling with Uncertain Processing Times

Sawat Pararach
Industrial Engineering Department, Engineering Faculty,
Thammasat University, Pathum Thani 12121.
e-mail: psawat@engr.tu.ac.th

Pichit Sukcharoenpong and Peerayuth Charnsethikul
Industrial Engineering Department, Engineering Faculty,
Kasetsart University, Bangkok 10900. Thailand.

Abstract

The scheduling classified as an NP-completed problem is challenging for many researchers. This paper is concerned with a flowshop scheduling problem with uncertain processing time. The model consists of a set of machines (multiple machines) and a set of jobs in a flowshop plant. The uncertain processing times can be represented by a discrete probability distribution. The objective is to find a job sequence such that the expected makespan is minimized. The existing lower bound requires large computational time. This paper proposes various lower bounds based on machine and job considerations. The experiments are conducted and compared with Barasubramanian and Grossman's algorithm. The results found that on average, the branch and bound using various lower bounds, used more branching nodes than the reference one, but required less computational time.

Keywords: Flowshop, Uncertain Processing Times, Branch and Bound.

1. Introduction

Scheduling is important in the production planning stage of manufacturing. It aims to help production planners and production managers to construct a time table of related tasks. A good scheduling helps manufacturers in reducing production time and cost. The Flowshop problem concerns the sequencing of a given number of jobs through a series of machines in exactly the same order on all machines, with the aim to satisfy a set of constraints and optimize a certain objective. The most studied objectives are frequently defined as makespan, meaning the completion time of the last job on the last machine. A large number of deterministic scheduling algorithms have been proposed in the last five decades [1]. Several objective functions for flowshop scheduling are considered, such as flowshop problems with separable setup time, which could be either sequence independent[2] or sequence dependent[3], a non-preemptive multiprocessor flowshop problem[4], a minimum weight combination of job flowtime and schedule makespan problem[5], a flowshop problem with minimum holding costs of inventory [6], and a hybrid two-stage flowshop with a batch processor in stage 1 and single processor in stage 2 [7]. In reality, the environment can be uncertain in a number of factors such as processing times and cost. As a result, a number of papers in recent years have addressed scheduling in the face of uncertainties under different parameters [8]. A solution to the flowshop scheduling problem arranges the order in which jobs are processed on each machine. In basic, there are many possible schedules and a common objective is to select the one that minimizes the completion time of a set of jobs. However, in the case of uncertain processing time, the solutions generated could not use the ordinary deterministic models.

For the flowshop given uncertain processing time, the objective is to find a jobs sequencing that minimizes the expected makespan. Barasubramanian and Grossmann considered the uncertainties in processing time using discrete probability distributions. The discrete probability distribution leads to a combinatorial explosion of the involved state space. They proposed a novel and rigorous branch and bound procedure for selecting the sequence with minimum expected makespan. However, calculation of the lower bound of unscheduled jobs in the branch and bound process requires permutation for the optimal value. This problem issue is needed to be addressed extensively. Then, the previous lower bound will be modified for a better efficiency.

The paper is begun with introducing the problem statement, the review of the lower bound of Balasubramanian and Grossman's algorithm. Then, the various lower bounds based on machines and jobs considerations are proposed. The last section deals with evaluation of the related algorithms and experimental comparisons.

2. Problem Statement

2.1 Flowshop with probabilistic processing time

In the flowshop environment, a set of n jobs must be scheduled on a set of m machines, where each job has the same routing of machines. More recent works present a sufficient condition on the processing time distributions that stochastically minimize the expected makespan in the case of a two-machine and a three-machine flowshop with unlimited intermediate storage, respectively. Given jobs, $j = 1,.., N$, that are to be produced in a flowshop plant with $i=1,..,M$ machines. Each of these jobs requires processing in all of the M machines and follows the same sequence of processing. All the machines have one processing unit each. The processing time of job j on machine i is a random variable, $T_{i,j}$. A discrete probability density function to describe the uncertainty in the processing times is shown in Figure 1.

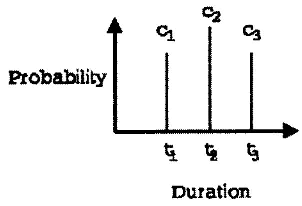

Figure 1. Duration uncertainty [3]

Figure 1 shows the discrete probability function, where C_k denotes the probability of a particular realization, k, t_k denote duration of a particular realization k, and the number of realizations is $|K| = 3$. A scenario is a combination of the realizations of all random variables. If there are four mutually independent random variables and each with $|K| = 3$ possible values, then, the total number of scenarios is 3^4.

2.2 Objective function

The problem is to determine a schedule from all possible permutations of the jobs for the flowshop plant with minimized expected makespan. The expected makespan is the average of the sum of all products of probability and completion time in each scenario.

2.3 Branch and Bound

The feasible set of solutions of the flowshop with probability processing time problem can be represented by a set of $N!$ sequence orders. An optimal solution can be obtained by the straightforward complete enumeration method and selecting the one with minimum objective value. However, a complete enumeration is hardly practical because the number of cases to be considered is usually enormous. A branch and bound method consists of general characteristics, branching rules, lower bounding rules, and search strategies. Balasubramanian and Grossmann [9] proposed a branch and bound method with a disaggregation algorithm. They disaggregate all the uncertainties associated with the tasks that are involved in the processing of scheduled jobs. They generate all possible scheduled jobs beginning and evaluate their expected makespans, using mean values for the processing times of the remaining jobs, and select the minimum value to get the lower bound. The lower bound must be obtained as a sequence of monotonically increasing lower bounds, ie. . $Z_A \le Z_{AB} \le Z_{ABC}$ if Z_A is the lower bound of job A located at the beginning of a work sequence. The branching rule discussed next is the first search depth and in the initial stage is combined with the local best solutions search to obtain an upper bound quickly. The nodes can be pruned if their lower bound is higher than the current upper bound.

2.4 Branch and Bound procedures[9]

1. *Initialization.* Denote the problem to be solved as P. Define the problem parameters N, M and the discrete probability space. Set $UB=+\infty$.

2. *Create the root node.* Apply algorithm *evaluate* (as below) to obtain the lower bound at the root node. Add root node to the list L of nodes yet to be fathomed.

3. *Selecting the subproblem.* If L is empty, i.e. $L = \varnothing$, there are no more subproblems to be solved, go to step 5. Else, choose a problem P_i (with value Z_i) from L and set $L = L / P_i$, go to step 5.

4. *Solving the subproblems.* Create the children, $P_i^1, P_i^2, \ldots$ of subproblem P_i and *evaluate* all of them. Let the values obtained be Z_i^1, Z_i^2, ... If all of the jobs have been fixed and all the uncertainties disaggregated in the children, then UB = min (UB, Z_i^1, Z_i^2, ...). Add to L the children whose values are not greater than the current UB. Go to step 3.

5. *Termination.* An optimal solution has been detected with the value UB. Stop.

Algorithm *evaluate*

1. *Initialization*, Initialize $LB = +\infty$. Let a partial sequence be fixed at this node and let this be denoted by SF.

- Aggregate all the uncertainties associated with jobs in the form of mean values of processing times, i.e. $\forall_{i,j}$, set:

$$\bar{T}_{ij} = \sum_{k=1}^{K} p_{ijk} t_{ijk}$$

2. *Generate sequence.* Generate sequence S by appending the jobs not present in SF to SF.

3. *Evaluate sequence.* Compute the expected value of the makespan of sequence S. Denote this value by Z_S. Update LB if Z_s is smaller than or equal to LB. Go to step 2.

4. *Termination.* Return LB as the lower bound at this node. Stop.

2.5 Balasubramanian and Grossmann's lower bound (B2002)

The lowerbound obtained from the algorithm of Balasubramanian and Grossman [9] that is described in this section is based on the permutation procedure that the worst case performance gives $N!$. It is defined as B2006. The proposed lowerbound that is described in section 3 is based on the Ignall-Shrage algorithm that the worst case performance gives 2^N.

The makespan (MS) equals total processing time on 1^{st} order on all machines + (completion time of 2^{nd} completion time of 1^{st}) + (completion time of 3^{rd} completion time of 2^{nd}) +,..,+ (completion time of N^{th} completion time of $(N\text{-}1)^{th}$). Mathematically,

$$MS = \sum_{j=1}^{M} T_{\sigma(1),j} + t_{\sigma(2),M} - t_{\sigma(1),M} + t_{\sigma(3),M} - t_{\sigma(2),M} + \ldots$$
$$\ldots, + t_{\sigma(N),M} - t_{\sigma(N-1),M} \quad (1)$$

where $T_{\sigma(1),j}$ is the processing time of the 1^{st} job on machine j and $t_{\sigma(i),j}$ is the completion time of the i^{th} job on machine j and σ denote the set of initial partial sequence.

In case of processing time uncertainties, the expected makespan (EMS) equals:

$$EMS = E[\sum_{j=1}^{M} t_{\sigma(1),j} + t_{\sigma(2),M} - t_{\sigma(1),M} +, \ldots$$
$$\ldots., + t_{\sigma(N),M} - t_{\sigma(N-1),M}]$$

From the above relationship, the expected value of the sum of random variables is the same as the sum of expected values of the random variables.

$$E[A+B-C] = E[A]+E[B]-E[C]$$

Then,

$$E[\sum_{j=1}^{M} t_{\sigma(1),j} + t_{\sigma(2),M} - t_{\sigma(1),M} + t_{\sigma(3),M} - t_{\sigma(2),M} +,...$$
$$..,+t_{\sigma(N),M} - t_{\sigma(N-1),M}]$$
$$= E[\sum_{j=1}^{M} T_{\sigma(1),j}] + E[t_{\sigma(2),M} - t_{\sigma(1),M}] +$$
$$E[t_{\sigma(3),M} - t_{\sigma(2),M}] +,...,$$
$$+E[t_{\sigma(N),M} - t_{\sigma(N-1),M}] \quad (2)$$

Property 1, for a given partial schedule σ, if the uncertain processing times are replaced by their mean values, a lower bound on the expected makespan of the given schedule is obtained.

So,

$$\sum_{j=1}^{M} \overline{T}_{\sigma(1),j} \leq E[\sum_{j=1}^{M} T_{\sigma(1),j}] \quad (3)$$

and:

$$\overline{t}_{\sigma(a),M} - \overline{t}_{\sigma(a-1),M} \leq E[t_{\sigma(a),M} - t_{\sigma(a-1),M}] \quad (4)$$

where $\overline{T}_{\sigma(1),j}$ is the expected processing time of the 1[st] job on machine j replaced by related mean values and $\overline{t}_{\sigma(i),j}$ is the expected completion time of the i^{th} job on machine j replaced by involved mean values. Branch and bound steps based on (3) and (4) requires calculation of the expected total processing time of a job which is scheduled in the 1^{st} position, $E[\sum_{j=1}^{M} T_{\sigma(1),j}] = \sum_{j=1}^{M} \overline{T}_{\sigma(1),j}$. For the job which is scheduled in the a^{th} position, calculate the expected completion time by: $E[t_{\sigma(a),M} - t_{\sigma(a-1),M}] = \overline{t}_{\sigma(a),M} - \overline{t}_{\sigma(a-1),M}$. For the unscheduled jobs, calculate the average completion time by using: $E[t_{\sigma(b),M} - t_{\sigma(b-1),M}] = \overline{t}_{\sigma(b),M} - \overline{t}_{\sigma(b-1),M}$.

For example, if jobs 1,.., h are scheduled jobs and jobs $h+1$,..,N are unscheduled jobs, Balasubramanian and Grossmann's lower bound can be obtained as (5).

$$LB = \overline{t}_{\sigma(h),M} + \min_{\sigma(h+1),...,\sigma(N)} \{\sum_{b=h+1}^{N} (\overline{t}_{\sigma(b),M} - \overline{t}_{\sigma(b-1),M})\} \quad (5)$$

where

$$\overline{t}_{\sigma(h),M} = \sum_{j=1}^{M} \overline{T}_{\sigma(1),j} + \sum_{a=1}^{h} (\overline{t}_{\sigma(a),M} - \overline{t}_{\sigma(a-1),M})$$

and $\min_{\sigma(h+1),...,\sigma(N)} \{\sum_{b=h+1}^{N} (\overline{t}_{\sigma(b),M} - \overline{t}_{\sigma(b-1),M})\}$ obtained from generation of all possible permutations on $\sigma(h+1)$, ..., $\sigma(N)$ (unscheduled jobs set) and choose the minimum value of:

$$\sum_{b=h+1}^{N} (\overline{t}_{\sigma(b),M} - \overline{t}_{\sigma(b-1),M}).$$

3. The proposed lower bounds

In the regular flowshop scheduling problems with constant processing times, a branch-and--bound procedure proposed by Ignall-Schrage [10] has a worst case performance analysis better than a permutation approach. It is based on two improving algorithms, one is a machine based lower bound approach[10] defined as P2006, the other is a job based lower bound approach[11] defined as S2006. The method that selects the best lower bound between P2006 and S2006 in each step is defined as M2006.

For the flowshop scheduling problems with uncertain processing times, Balasubramanian and Grossmann's lower bound[9] considered a lower bound based on a permutation procedure. This section considers two lower bound algorithms based on a machine based lower bound approach[10] and a job based lower bound approach[11].

Let σ denote the set of initial partial sequence in which jobs in the first h positions have been fixed, σ (j) = *([1],..,[h])*. Let σ' denote the set of unscheduled jobs, σ'(j) = *([h+1],..,[N])* and $\sigma + \sigma' = N$.

3.1 Machine based lower bound (P2006)

A Machine based lower bound originated from Ignall & Schrage in 1965 [10]. The basic concept is to compute the total expected completion time without idle time on a machine referred to as the critical path or the bottleneck machine. The lower bound is:

$$LB^{M}_{\sigma(h)} = \max_{1\leq k\leq M} \{\overline{t}_{\sigma(h),k} + \sum_{b=h+1}^{N} \overline{T}_{\sigma(b),k} + \min_{h+1\leq r\leq N} \{\sum_{g=k+1}^{M} \overline{T}_{\sigma(r),g}\}\} \quad (6)$$

where $\sum_{b=h+1}^{N} \bar{T}_{\sigma(b),k} + \min_{h+1\le r\le N}\{\sum_{g=k+1}^{M} \bar{T}_{\sigma(r),g}\}$ is the total expected completion time on machine k for job $\sigma(h+1), \ldots, \sigma(N)$ which are unscheduled jobs. Then, the maximum total expected completion time on a machine is selected.

Proposition 1 The proposed machine based lower bound is lower than Balasubramanian and Grossmann 's by the following.

$$\max_{1\le k\le M}\{\bar{t}_{\sigma(h),k} + \sum_{b=h+1}^{N} \bar{T}_{\sigma(b),k} + \min_{h+1\le r\le N}\{\sum_{g=k+1}^{M} \bar{T}_{\sigma(r),g}\}\} \le$$

$$\bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\} \quad (7)$$

Proof

Given v as a machine with the maximum lower bound in (7). Then:

$$\bar{t}_{\sigma(h),v} + \min_{h+1\le r\le N}\{\sum_{g=v+1}^{M} \bar{T}_{\sigma(r),g}\} + \sum_{b=h+1}^{N} \bar{T}_{\sigma(b),v}$$

$$\le$$

$$\bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\} \quad (8)$$

If $\bar{I}_{\sigma(h),g}$ is the expected waiting time of the h^{th} job on machine g, then:

$$\bar{t}_{\sigma(h),g} = \bar{I}_{\sigma(h),g} + \bar{T}_{\sigma(h),g} \quad (9)$$

However, the expected completion time on machine M, is:

$$\bar{t}_{\sigma(h),M} = \bar{t}_{\sigma(h),v} + \sum_{g=v+1}^{M}(\bar{I}_{\sigma(h),g} + \bar{T}_{\sigma(h),g}) \quad (10)$$

$$\bar{t}_{\sigma(h),v} + \min_{h+1\le r\le N}\{\sum_{g=v+1}^{M} \bar{T}_{\sigma(r),g}\} \le \bar{t}_{\sigma(h),M} \quad (11)$$

since,

$$\bar{t}_{\sigma(b),v} - \bar{t}_{\sigma(b-1),v} = \bar{I}_{\sigma(b),v} + \bar{T}_{\sigma(b),v}$$

then,

$$\bar{t}_{\sigma(b),v} - \bar{t}_{\sigma(b-1),v} \ge \bar{T}_{\sigma(b),v} \quad (12)$$

from (10) and (11),

$$\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),v} - \bar{t}_{\sigma(b-1),v}) \le \sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M}) \quad (13)$$

From (11) and (13), it can be concluded that (8) is true. □

Proposition 2 The Machine based lower bound in (8) can be calculated with a less maximum number of steps than the reference lower bound.

Proof

The lower bound $\max_{1\le k\le M}\{\bar{t}_{\sigma(h),k} + \min_{h+1\le r\le N}\{\sum_{g=k+1}^{M} \bar{T}_{\sigma(r),g}\} + \sum_{b=h+1}^{N} \bar{T}_{\sigma(b),k}\}$ can be calculated within $M*(1+N^2M)$ steps,

whereas,

$$\bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\}$$

can be calculated within $(N-1)!$, therefore, the proposition holds. □

3.2 Job based lower bound (S2006)

A Job based lower bound is the extension of Mahon & Burton firstly proposed in 1967 [11]. The lower bound considers the total expected processing time of the critical path based on jobs as:

$$LB^{L}_{\sigma(h)} =$$

$$\max_{1\le k\le M}\{\bar{t}_{\sigma(h),k} + \max_{h+1\le i\le N}\left[\sum_{g=k}^{M}\bar{T}_{\sigma(i),g} + \sum_{\substack{b=h+1\\ b\ne i}}^{N}\min(\bar{T}_{\sigma(b),k},\bar{T}_{\sigma(b),M})\right]\} \quad (14)$$

where $\max_{h+1\le i\le N}\left[\sum_{g=k}^{M}\bar{T}_{\sigma(i),g} + \sum_{\substack{b=h-1\\ b\ne i}}^{N}\min(\bar{T}_{\sigma(b),k},\bar{T}_{\sigma(b),M})\right]$ is the maximum value of the sum of the total of the expected processing time of the i^{th} job under unscheduled job $\sigma(h+1, \ldots, \sigma(N)$ processed on machine k up to M and the minimum expected processing time remaining on machine k or M of remaining unscheduled jobs.

Proposition 3 The job based lower bound value is lower than the reference lower bound or:

$$\max_{1\le k\le M}\{\bar{t}_{\sigma(h),k} + \max_{h-1\le i\le N}\left[\sum_{g=k}^{M}\bar{T}_{\sigma(i),g} + \sum_{b=h+1,b\ne i}^{N}\min\left(\bar{T}_{\sigma(b),k},\bar{T}_{\sigma(b),M}\right)\right]$$
$$\le \bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\} \quad (15)$$

Proof
Given v and u as the machine and the job from (15) is maximized, then:

$$\bar{t}_{\sigma(h),v} + \sum_{g=k}^{M}\bar{T}_{\sigma(u),g} + \sum_{b=h+1,b\ne u}^{N}\min\left(\bar{T}_{\sigma(b),v},\bar{T}_{\sigma(b),M}\right)$$
$$\le \bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\} \quad (16)$$

Swap the u^{th} job to position h+1, then:

$$\bar{t}_{\sigma(h),v} + \sum_{g=k}^{M}\bar{T}_{\sigma(h+1),g} + \sum_{b=h+2}^{N}\min\left(\bar{T}_{\sigma(b),v},\bar{T}_{\sigma(b),M}\right)$$
$$\le \bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\} \quad (17)$$

The fact that,

$$\bar{t}_{\sigma(h),v} + \sum_{g=k}^{M}\bar{T}_{\sigma(h+1),g} \le \bar{t}_{\sigma(h+1),M} \quad (18)$$

and:

$$\sum_{b=h+2}^{N}\min\left(\bar{T}_{\sigma(b),v},\bar{T}_{\sigma(b),M}\right) \le \bar{t}_{\sigma(N),M} - \bar{t}_{\sigma(h+1),M} \quad (19)$$

By combining (18) and (19) then,

$$\bar{t}_{\sigma(h),v} + \sum_{g=k}^{M}\bar{T}_{\sigma(h+1),g} + \sum_{b=h+2}^{N}\min\left(\bar{T}_{\sigma(b),v},\bar{T}_{\sigma(b),M}\right) \le \bar{t}_{\sigma(N),M} \quad (20)$$

and from:

$$\bar{t}_{\sigma(N),M} \ge \bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\}$$

and $\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M} = \bar{I}_{\sigma(b),M} + \bar{T}_{\sigma(b),M}$

then:

$$\bar{t}_{\sigma(h),v} + \sum_{g=k}^{M}\bar{T}_{\sigma(h+1),g} + \sum_{b=h+2}^{N}\min\left(\bar{T}_{\sigma(b),v},\bar{T}_{\sigma(b),M}\right)$$
$$\le \bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\} \quad (21)$$

Therefore, the proposition holds as stated.

Proposition 4 The job based lower bound can be calculated with a less maximum number of steps than the reference lower bound.

Proof Since the lower bound using:

$$\max_{1\le k<M}\{\bar{t}_{\sigma(h),k} + \max_{h+1<i\le N}\left[\sum_{g=k}^{M}\bar{T}_{\sigma(i),g} + \sum_{b=h+1,b\ne i}^{N}\min\left(\bar{T}_{\sigma(b),k},\bar{T}_{\sigma(b),M}\right)\right]$$

requires the maximum number of steps in calculations of M*(1+N^2M) as a polynomial increasing function of the numbers of job and machine, whereas:

$\bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\}$ can be calculated with the number of steps up to (N-1)!, an exponential increasing function. Hence, the proposition holds.

3.3 Composite based lower bound (M2006)

The maximum value of lower bound in (6) and (14) is the composite based lower bound.

$$GBL_{\sigma(h)} = \max\{LB^{M}_{\sigma(h)}, LB^{L}_{\sigma(h)}\} \quad (22)$$

Proposition 5 The composite based lower bound (23) is lower than the reference lower bound.

$$GBL_{\sigma(h)} = \max\{LB^{M}_{\sigma(h)}, LB^{L}_{\sigma(h)}\} \le$$
$$\bar{t}_{\sigma(h),M} + \min_{\sigma(h+1),\ldots,\sigma(N)}\{\sum_{b=h+1}^{N}(\bar{t}_{\sigma(b),M} - \bar{t}_{\sigma(b-1),M})\} \quad (23)$$

Obviously, the proposition holds as stated.

4. Results and Discussion

The algorithms in the previous section are implemented on a personal computer with CPU Pentium 4 and 256 MB RAM. The results can be illustrated in Table 1 to 4.

Table 1 Average CPU time for each problem size ($|K| \leq 3$)

Job s	M/C	No. of scenarios	Max.Pro c. time	Expected Makespan	CPU time (Seconds)			
					B2002	P2006	S2006	M2006
6	5	216	100	560.6650	0.0520	0.0523	0.0520	0.0517
		432		521.3320	0.2603	0.2550	0.2603	0.2603
		648		495.6640	0.1560	0.1613	0.1510	0.1560
		864		560.3307	0.1303	0.1350	0.0940	0.1353
		1296		498.3290	0.2813	0.2657	0.2863	0.2863
		1728		530.9983	0.4063	0.4360	0.3957	0.3853
		2592		490.3393	0.4320	0.3957	0.4010	0.4063
7	5	648	100	586.9950	0.5260	0.5047	0.5213	0.5260
8				574.3300	0.2607	0.2397	0.2343	0.2397
9				534.9963	1.3073	1.1200	1.1360	1.1720
10				597.7100	2.4589	1.4457	1.4994	1.4823
11				743.3300	97.6613	76.5310	77.6510	77.7137
12				676.9927	688.4690	447.2797	433.7134	439.0270
13				759.9930	8,380.4700	6,331.44	6,315.1100	6,450.8600

Table 2 Average number of nodes for each problem size ($|K| \leq 3$)

Jobs	M/C	No. of scenarios	Max.Proc. time	Expected Makespan	Number of branching nodes			
					B2002	P2006	S2006	M2006
6	5	216	100	560.6650	19.00	76.33	28.67	26.00
		432		521.3320	60.67	103.333	93.6667	78.3
		648		495.6640	38.00	45.000	41.3333	38.7
		864		560.3307	36.33	76.000	42.3333	54.3
		1296		498.3290	61.67	115.000	95.0000	77.3
		1728		530.9983	67.33	106.000	107.0000	80.7
		2592		490.3393	54.67	79.333	72.0000	62.7
7	5	648	100	586.9950	150.00	295.00	676.67	223.67
8				574.3300	74.00	150.33	395.67	109.33
9				534.9963	303.00	798.00	9,120.67	458.67
10				597.7100	460.14	1,113.67	16,628.24	774.67
11				743.3300	16,899.00	240,281.33	325,490.33	92,486.67
12				676.9927	754,84.33	402,015.33	183,450.33	138,434.0
13				759.993	1,005,310.0	1,114,190.0	1,017,060.0	1,013,010.0

Table 1-4 All time is Seconds.

Table 3 Average CPU time for each problem size ($|K| \leq 3$)

Jobs	M/C	No. of scenarios	Max.Proc. time	Expected Makespan	CPU time (Seconds)			
					B2002	P2006	S2006	M2006
6	3	864	10	46.66	0.5103	0.5103	0.5387	0.5417
			40	158.67	0.1663	0.1560	0.1613	0.1613
			70	288.33	0.2553	0.2760	0.2710	0.2703
			100	455.33	0.3387	0.3440	0.3440	0.3490
			130	551.33	0.1297	0.1353	0.1303	0.1353
4	4	648	100	367.9983	0.0623	0.0680	0.0727	0.0677
	6			562.3293	0.0210	0.0260	0.0263	0.0260
	8			626.9963	0.0257	0.0260	0.0263	0.0317
	10			752.6623	0.0627	0.0523	0.0573	0.0573
	12			796.9993	0.0727	0.0727	0.0783	0.0710

Table 4 Average number of nodes for each problem size ($|K| \leq 3$)

Jobs	M/C	No. of scenarios	Max.Proc. time	Expected Makespan	Number of branching nodes			
					B2002	P2006	S2006	M2006
			10	46.66	114.67	117.33	116.0	116.0
			40	158.67	51.0	77.00	73.00	56.7
6	3	864	70	288.33	69.67	80.0	69.67	69.7
			100	455.33	86.67	97.0	89.67	87.3
			130	551.33	51.00	100.7	178.67	68.0
	4			367.9983	17.33	19.67	18.33	17.3
	6			562.3293	6.67	10.67	9.00	8.7
4	8	648	100	626.9963	8.00	12.00	10.33	10.3
	10			752.6623	13.00	16.33	15.00	14.0
	12			796.9993	14.33	17.33	16.00	15.7

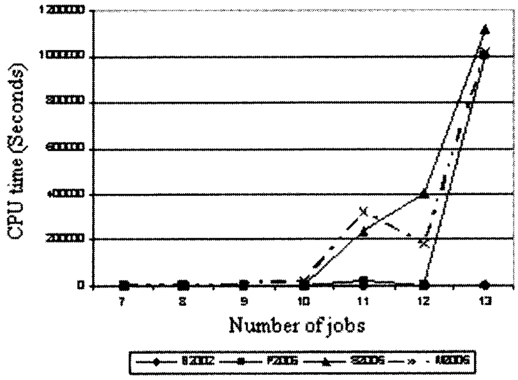

Figure 2 CPU time and number of jobs

Graphically, Figure 2 shows how N affects CPU time. The variation of CPU time with respect to M is shown in Figure 3. It is clear from Table 1 and Figure 2 that for increasing of N, CPU time increases exponentially while increases M, (Table 3 and Figure 3), do not clearly affect CPU time. For given N=6, (Table 2 and Figure 4), increases number of scenarios, increases CPU time. It is clear from Table 2 and Figure 5 that for increases of N, the number of branching nodes increases.

In case of N = 6, It was found from Table 2 and Figure 4 that the number of total scenarios influences the CPU time while the number of machines, (Table 4 and Figure 7), does not influence the number of nodes. The maximum processing times do not influence the CPU time and the number of nodes, (Table 3, Table 4 and Figure 6). The number of jobs which can be solved on a branch and bound algorithm within a reasonable time is no more than N = 13. Larger problems, however, can be solved by approximation of existing exact solutions and will be worth further investigation.

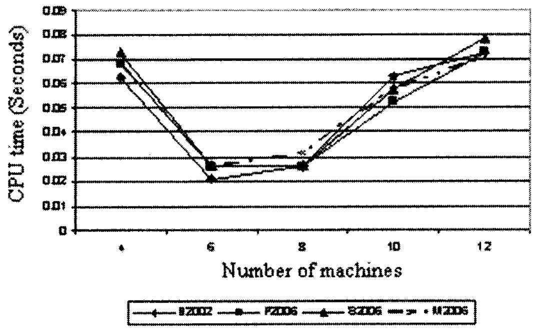

Figure 3 Number of machines and CPU time

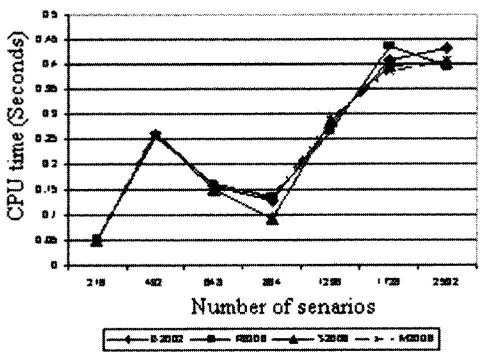

Figure 4 Number of scenarios and CPU time

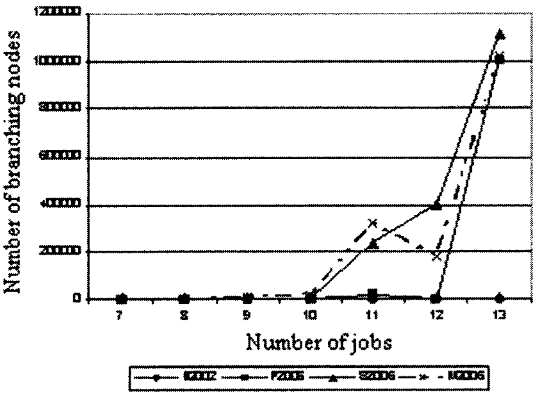

Figure 5 Number of branching nodes and number of jobs

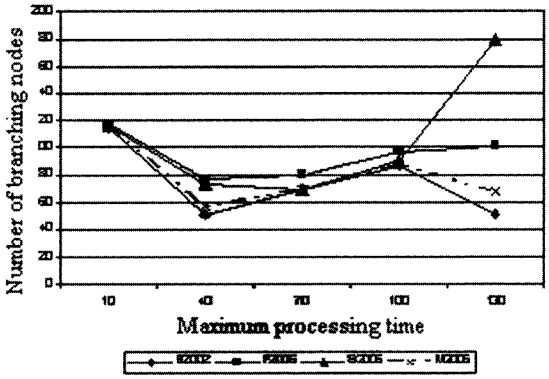

Figure 6 Maximum processing time and number of branching nodes

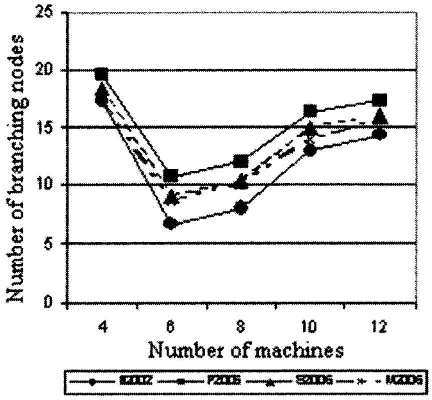

Figure 7 Number of machines and number of branching nodes

5. Conclusion

The flowshop scheduling problem with uncertain processing time is studied, because of the real case that the processing times are uncertain due to uncontrollable factors. The study assumed that each processing time could be defined as a discrete probability function. The objective of the problem is to find a job schedule that minimizes the expected makespan. The proposed lower bounds have a worst performance analysis of less than the original procedure by Propositions 1-5. The experimental results show that the proposed lower bounds are faster than the original of Balasubramanian and Grossman [3]. But they are weaker than the original one in term of the number of branching nodes. For future research, the performance evaluating of the composite and the Job based lower bound are required for a better result.

6. Acknowledgement

The author gratefully acknowledges financial support from the Ministry of University and Thammasat University, Thailand.

Thanks to T. Aramgiatsiris for his advice on programming.

7. References

[1] Gupta J.N.D., and Stafford E.F., Flowshop Scheduling Research After five Decades, European Journal of Operational Research, Vol. 169, No 3, pp.669-711, 2006.

[2] Allahverdi, A., Gupta, J.N.D., and Aldowaisan, T., A Survey of Scheduling Research Involving Setup Considerations, OMEGA, International Journal of Management Science 27, pp.219-239, 1999.

[3] Rubén R., Concepción, M., and Alcaraz, J., Solving the Flowshop Scheduling Problem with Sequence Dependent Set up Times using Advanced Metaheuristics, European Journal of Operational Research, Vol. 165, pp.34-54, 2005.

[4] Tamás K., amd Erwin, P., A Review of Exact Solution Methods for the Non-preemptive Multiprocessor Flowshop Problem, European Journal of Operational Research, Vol. 164, pp.592-608, 2005.

[5] Wei-Chang Yeh, A New Branch-and-bound Approach for the $n/2/\text{Flowshop}/\alpha F+\beta C_{max}$

Scheduling Problem, Computer & Operations Research, Vol. 26, pp. 1293-1310, 1999.

[6] Moon-Wong Park and Yeong-Dae Kim, A Branch and Bound Algorithm for a Product Scheduling Problem in an Assembly System Under Due Date Constraints, European Journal of Operational Research, Vol. 123, pp.504-518, 2000.

[7] Ling-Heuy Su, A Hybrid Two-stage Flowshop with Limited Waiting Constraints, Computers & Industrial Engineering, Vol. 44, pp.409-424, 2003.

[8] Kamburowski J., Stochastically Minimizing the Makespan in Two-machine Flow Shops Without Blocking, European Journal of Operational Research, Vol. 112, pp. 304-309, 1999.

[9] Barasubramanian J., and Grossman I.E., A Novel Branch and Bound Algorithm for Scheduling Flowshop Plants with Uncertain Processing Times, Computers & Chemical Engineering, Vol. 26, pp.46-57, 2002.

[10] Ignall E. and Schrage L.E., Application of the Branch and Bound Technique to Some Flow–shop Scheduling Problems, Operations Research, Vol.13, pp. 400-412, 1965.

[11] Mahon G.M.B. and Burton P.G., Flow-shop Scheduling with the Branch and Bound Method, Operations Research, Vol.15, No.3, pp. 473-481, 1967.