Journal of mathematics and computer Science 6 (2013), 238-250

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

Stochastic Online Scheduling With Preemption Penalties

Mehdi heydari, Mohammad Mahdavi Mazdeh and Mohammad Bayat[+]

Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Mheydari@iust.ac.ir, mazdeh@iust.ac.ir, mohammadbayat@iust.ac.ir

Article history:
Received February 2013
Accepted April 2013
Available online April 2013

Abstract

This paper considers a stochastic online scheduling problem in which a set of independent jobs are to be processed on a single machine. Each job has a processing time, which is a random variable with normal distribution. All the jobs arrive overtime, which means that the existence and the parameters of each job including its processing time specifications and weight are unknown until its release date. Moreover, the actual processing time of each job is unknown until its completion. During the processing, jobs are allowed to be preempted and restarted later. So, the processing time devoted to the job before the preemption is lost and considered as preemption penalty. The objective is to minimize the expected value of the total weighted completion time. Since the problem is strongly NP-hard, a heuristic algorithm is proposed in this paper and is validated using numerical examples. The proposed method utilizes the properties of the normal distribution but it can be used as a heuristic for other distributions, as long as their means and variances are available.

Keywords: Stochastic scheduling, online scheduling, preemption penalty, job preemption, preemption-restart

1. Introduction

In stochastic scheduling, the processing time is uncertain and we don’t have exact information about it. The problem will be more complicated when jobs arrive online, or in other words, jobs and their characteristics are unknown before arriving to the shop. In addition, any job can be preempted at any time. In the previous stochastic researches, preemption is usually assumed to be ‘free’ which means that each job can be preempted at any point of time and resume later without a penalty. However, this is usually not true in practice. In many cases such as melting furnaces, all the operations performed

[+] Corresponding author

Tel.: +98--9126971409; P.O. Box 16846-13114, Tehran, Iran; E-mail address: Mohammadbayat@iust.ac.ir

before the preemption are considered lost (preemption-restart model). Thus, the time that has been spent on the job before the preemption is considered as preemption penalty.

Several papers have studied online scheduling problem with preemption in deterministic mode. Potts and Wassenhove [1] proposed considering preemption penalties under the lot-sizing model. Liu and Cheng [2] considered a single-machine scheduling problem with job-independent setup times and used a greedy algorithm for minimizing the total completion time. Zheng et al. [3] proposed an algorithm for the unit length and general length cases for online scheduling in preemption-restart model with preemption penalties. Zheng et al. [4] considered a case that the preemption penalty is equal to value of the job. Stanley P.Y. Fung [5] considered online deadline scheduling problem with preemption and proposed lower bounds for it. Megow [16] considered the shortest weighted remaining processing time rule (*SWRPT*) in which at any moment the job with largest ratio of weight over remaining processing time is processed. Heydari et al. [6] proposed a heuristic algorithm to minimize the total flow time in single machine online scheduling with preemption penalties. Heyadi and Mohammadi [17]#considered the single-machine scheduling problem of minimizing the total flow time subject to job release dates and fuzzy preemption penalties.

Some other papers have considered preemptive stochastic scheduling in offline mode. These studies include Chazan et al [7], Konheim [8] and Sevcik [9]. Weiss [10] proposed a rule, namely Gittins index priority policy (*GIPP*) for preemptive stochastic scheduling problem $1|ptmn|E(\sum w_j c_j)$ which solves the problem optimally. Kämpke [11] investigated the problem $P|ptmn|E(\sum w_j c_j)$ with multiple machines, and proved that the *SEPT* (shortest expected processing time) policy is optimal when the weights are agreeable.

Few researches have addressed the issue of preemptive stochastic online scheduling. Megow and Vredeveld [12] considered stochastic online scheduling in preemption allowed mode and proposed two policies. These policies have been derived from *GIPP*. They also applied the policies for a problem with two parallel machines and proved that they are 2-approximative for preemptive stochastic online problem $P|r_j, ptmn|E[\sum w_j c_j]$. Manzhan and Xiwen [13] proposed the optimal policy, named SMPR, for the single machine preemptive stochastic scheduling problem, where jobs have a common arriving time, for minimizing the expected value of the total weighted completion time. They also introduced Uniform-Machine Priority Rule (UMPR), which is devised for the preemptive stochastic online scheduling on two uniform machines. Then, they proved that UMPR has an approximation factor of 2.

Therefore, most of the previous studies have considered preemption-resume mode and less attention has been paid to stochastic online scheduling with preemption penalties. We have not found a considerable study that performed to solve the problem of preemptive stochastic online scheduling in preemption-restart mode. Preemptive online stochastic scheduling problem $1|r_j, ptmn|E(\sum w_j c_j)$, is well known to be NP-hard, even in the case of equal release dates [14, 15].

The remainder of this paper is organized as follows: in Section 2, research assumptions and definitions have been introduced. Section 3 introduces problem in preemptive stochastic mode considering all of completed and uncompleted jobs. In Section 4, we generate priority rules and propose a heuristic. In section 5, a numerical example has been solved with proposed algorithm. The performance analysis of the proposed heuristic method is presented in section 6. Finally, the conclusions are discussed in the last part.

2. Research Assumptions and Definitions

Each job j has associated stochastic processing time t_j and release time r_j. The job is not available before its arrival and can be processed right on or after r_j. Parameter c_j denotes the completion time of job j . Moreover, we have no information about the job before r_j . The processing time is a random variable with normal distribution.

$$t_j \sim N(\mu_j, \sigma_j^2)$$

Which μ_j and σ_j^2 are mean and variance of processing. A machine can process at most one job at a time, and preemption is allowed for all jobs. The ultimate objective is to minimize expected value of total weighted completion time. Uncompleted jobs are scheduled at any time with weighted shortest expected processing time rule (*WSEPT*). This heuristic assigns the job with the largest ratio of weight to expected processing time, among the jobs that have arrived but haven't been completed yet.

3. Preemptive stochastic problem

Assume that during the processing of job x the job y arrives to the shop at r_y . The processing time of job x and y are random variables with normal distribution and following characteristics:

$$t_x \sim N(\mu_x, \sigma_x^2)$$

$$t_y \sim N(\mu_y, \sigma_y^2)$$

Considering the jobs before and after x and y , the status is demonstrated in Figure 1.

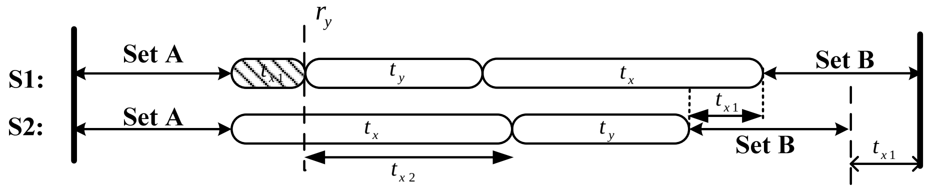

Figure 1. Schedules S_1 and S_2 in general mode

Where A is the subset of completed jobs and subset B is the uncompleted jobs. As shown in the Figure 1, Schedules S_1 and S_2 are either related to preemption or not preemption of job x at arrival of job y with regard to the situation of tasks before and after x . In both schedules, the distribution of total weighted completion time must be computed.

$$R = \sum f_1 = \sum_{\forall j \in A} f_{1j} + f_{1x} + f_{1y} + \sum_{\forall j \in B} f_{1j}$$

$$Q = \sum f_2 = \sum_{\forall j \in A} f_{2j} + f_{2x} + f_{2y} + \sum_{\forall j \in B} f_{2j}$$

Since the jobs before x have been completed, their processing time is definite. Therefore, their expected weighted completion time is constant value and equal in both schedules:

$$\sum_{\forall j \in A} f_{1j} = \sum_{\forall j \in A} f_{2j} = \lambda$$

The jobs in the set B have been arrived to the shop at a point of time before start time of job x , thus:

$$\forall j \in B \quad r_j \le s_x$$

Where s_x is the start time of job x . The job x is being processed and index j is used for it. We assume the number of jobs which are present in the shop is k and the number of uncompleted jobs is $k-j$. Also, assume that jobs are indexed with priority in set B as shown in Figure 2.

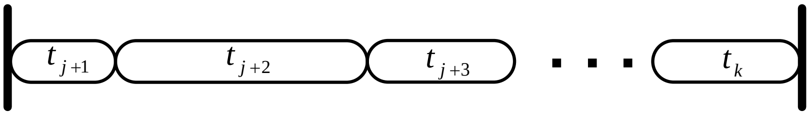

Figure 2. Jobs in set B

In schedule 1, total weighted completion time in set B is given by:

$$\sum_{\forall j \in B} f_{1j} = \sum_{h=j+1}^{k} f_{1h}$$

$$f_{1(j+1)} = w_{j+1}(s_x + t_{x1} + t_y + t_x + t_{j+1})$$

$$f_{1(j+2)} = w_{j+2}(s_x + t_{x1} + t_y + t_x + t_{j+1} + t_{j+2})$$

$$f_{1(j+3)} = w_{j+3}(s_x + t_{x1} + t_y + t_x + t_{j+1} + t_{j+2} + t_{j+3})$$

.

.

$$f_{1k} = w_k (s_x + t_{x1} + t_y + t_x + t_{j+1} + t_{j+2} + t_{j+3} + \ldots + t_k)$$

$$\Rightarrow E(\sum_{\forall j \in B} f_{1j}) = \sum_{h=j+1}^{k} E(f_{1h})$$

$$= \sum_{h=j+1}^{k} w_h (s_x + t_{x1} + \mu_y + \mu_x) + \sum_{h=j+1}^{k} w_h (\sum_{l=j+1}^{h} \mu_l)$$

Based on the above calculations to obtain the mean and variance of R we have:

$$R = \sum_{\forall j \in A} f_{1j} + f_{1x} + f_{1y} + \sum_{\forall j \in B} f_{1j}$$

$$= \lambda + w_x (s_x + t_{x1} + t_y + t_x) + w_y (s_x + t_{x1} + t_y) + \sum_{\forall j \in B} f_{1j}$$

$$E(R) = \lambda + (w_x + w_y)(s_x + t_{x1}) + \mu_y (w_x + w_y) + w_x \mu_x + \sum_{h=j+1}^{k} w_h (s_x + t_{x1} + \mu_y + \mu_x + (\sum_{l=j+1}^{h} \mu_l))$$

$$\mathrm{var}(R) = \sigma_y^2 (w_x^2 + w_y^2) + \sigma_x^2 w_x^2 + \sum_{h=j+1}^{k} w_h^2 (\sigma_x^2 + \sigma_y^2 + \sum_{l=j+1}^{h} \sigma_l^2)$$

Also mean and variance of $Q = \sum f_2$ are computed by:

$$E(Q) = \lambda + s_x (w_x + w_y) + \mu_x (w_x + w_y) + w_y \mu_y + \sum_{h=j+1}^{k} w_h (s_x + \mu_x + \mu_y + \sum_{l=j+1}^{h} \mu_l)$$

$$\mathrm{var}(Q) = \sigma_x^2 (w_x^2 + w_y^2) + \sigma_y^2 w_y^2 + \sum_{h=j+1}^{k} w_h^2 (\sigma_x^2 + \sigma_y^2 + \sum_{l=j+1}^{h} \sigma_l^2))$$

The preemption will occur if the total weighted completion time value in first schedule is less than in the second one. Let $G = R - Q$. Distribution function of G is as follows:

$$p(R<Q)=P(G<0)$$

$$E(G)=E(R)-E(Q)=(w_x+w_y)(t_{x1})+\mu_y w_x-\mu_x w_y+\sum_{h=j+1}^{k} w_h t_{x1}$$

$$=\mu_y w_x-\mu_x w_y+t_{x1}(w_x+w_y+\sum_{h=j+1}^{k} w_h)$$

$$\mathrm{var}(G)=\mathrm{var}(R)+\mathrm{var}(Q)$$

$$=\sigma_y^2(w_x^2+2w_y^2)+\sigma_x^2(2w_x^2+w_y^2)+\sum_{h=j+1}^{k} 2w_h^2(\sigma_x^2+\sigma_y^2+\sum_{l=j+1}^{h}\sigma_l^2))$$

$$\Rightarrow G\sim N(\mu_y w_x-\mu_x w_y+t_{x1}(w_x+w_y+\sum_{h=j+1}^{k} w_h),\sigma_y^2(w_x^2+2w_y^2)+\sigma_x^2(2w_x^2+w_y^2)+\sum_{h=j+1}^{k} 2w_h^2(\sigma_x^2+\sigma_y^2+\sum_{l=j+1}^{h}\sigma_l^2))$$

If the probability $p(G<0)$ is greater than the specified value of α then the preemption will occur, and otherwise the preemption is not allowed. The probability $p(G<0)$ is given by:

$$\xi=\frac{\mu_x w_y-\mu_y w_x-t_{x1}(w_x+w_y+\sum_{h=j+1}^{k} w_h)}{\sqrt{\sigma_y^2(w_x^2+2w_y^2)+\sigma_x^2(2w_x^2+w_y^2)+\sum_{h=j+1}^{k} 2w_h^2(\sigma_x^2+\sigma_y^2+\sum_{l=j+1}^{h}\sigma_l^2)}}$$

$$\Rightarrow p(G<0)=\Phi(\xi)$$

Where Φ is cumulative distribution function (cdf) of standard normal.

4. Proposed algorithm

In this section, we first extract rules for preemption based on the results of previous section.

Lemma 1- For $\alpha>0.5$, preemption will not occur if $\frac{\mu_x}{\mu_y}\le\frac{w_x}{w_y}$.

Proof: if $\frac{\mu_x}{\mu_y}\le\frac{w_x}{w_y}$ then $p(G<0)<\alpha$ and preemption will not occur because:

$$\frac{\mu_x}{\mu_y}\le\frac{w_x}{w_y}\rightarrow\mu_x w_y-\mu_y w_x\le 0$$

$$\rightarrow\mu_x w_y-\mu_y w_x-t_{x1}(w_x+w_y+\sum_{h=j+1}^{k} w_h)\le 0$$

$$\rightarrow\xi\le 0\xrightarrow{\alpha>0.5}\Phi(\xi)\le 0.5<\alpha$$

So, preemption is not allowed.

Lemma 2- If $\alpha>0.5$ then the necessary condition for the preemption of job x is

$$\mu_x w_y-\mu_y w_x>t_{x1}(w_x+w_y+\sum_{h=j+1}^{k} w_h).$$

Proof: If this condition is not satisfied we have:

$$\mu_x w_y - \mu_y w_x \le t_{x1}(w_x + w_y + \sum_{h=j+1}^{k} w_h)$$

$$\Rightarrow \mu_x w_y - \mu_y w_x - t_{x1}(w_x + w_y + \sum_{h=j+1}^{k} w_h) \le 0$$

$$\Rightarrow \xi \le 0 \Rightarrow \Phi(\xi) \le 0.5 < \alpha$$

Therefore, according to the proof of lemma 1, in this situation preemption will not occur.

Hence, a heuristic algorithm is proposed to minimize expected value of the total weighted completion time for single machine stochastic online scheduling problem in preemption-restart mode. In addition, a schematic exhibition of the algorithm is shown in Figure 3.

***Step* 0:** (parameter definition): Parameter k denotes the total number of jobs in the shop and j is the job counter parameter whit primary value of 1. Processing time of each job (t_j) is a random variable with normal distribution. Arrival time of job j is r_j. Parameter x denotes the job that is being processed and y denotes new job that arrives to the shop.

***Step* 1:** Prioritize all uncompleted jobs with *WSEPT* rule, from j to k.

***Step* 2:** Start the processing of job j and assign index x to its characteristics until it is completed or a new job arrives to the shop. If a new job arrives to the shop then assign index y to its characteristics and go to step 4, otherwise go to next step.

***Step* 3:** After completion of job x, if x was the last job then finish the algorithm else increment the counter j by one and go to step 1.

$j \leftarrow j + 1$

***Step* 4:** if $\alpha > 0.5$ then go to next step, otherwise go to step 7.

***Step* 5:** If $(\frac{\mu_x}{\mu_y} > \frac{w_x}{w_y})$ then follow the algorithm, otherwise go to step 9.

***Step* 6:** If $\mu_x w_y - \mu_y w_x > t_{x1}(w_x + w_y + \sum_{h=j+1}^{k} w_h)$ then follow the algorithm, otherwise go to step 9.

***Step* 7:** If $\Phi(\xi) \ge \alpha$ then follow the algorithm, otherwise go to step 9.

***Step* 8:** The priority of job y is more than job x. Thus, job x will be preempted and job y will be preferred. From this point of time, index j and parameter x will be used for job y. Increment the counter k by one, and then go to step 2.

$k \leftarrow k + 1$

***Step* 9:** Preemption is not allowed and the processing of job x will be continued. Prioritize newly arrived job among uncompleted jobs with *WSEPT* rule. Increment the counter k by one, and then go to step 2.

$k \leftarrow k + 1$

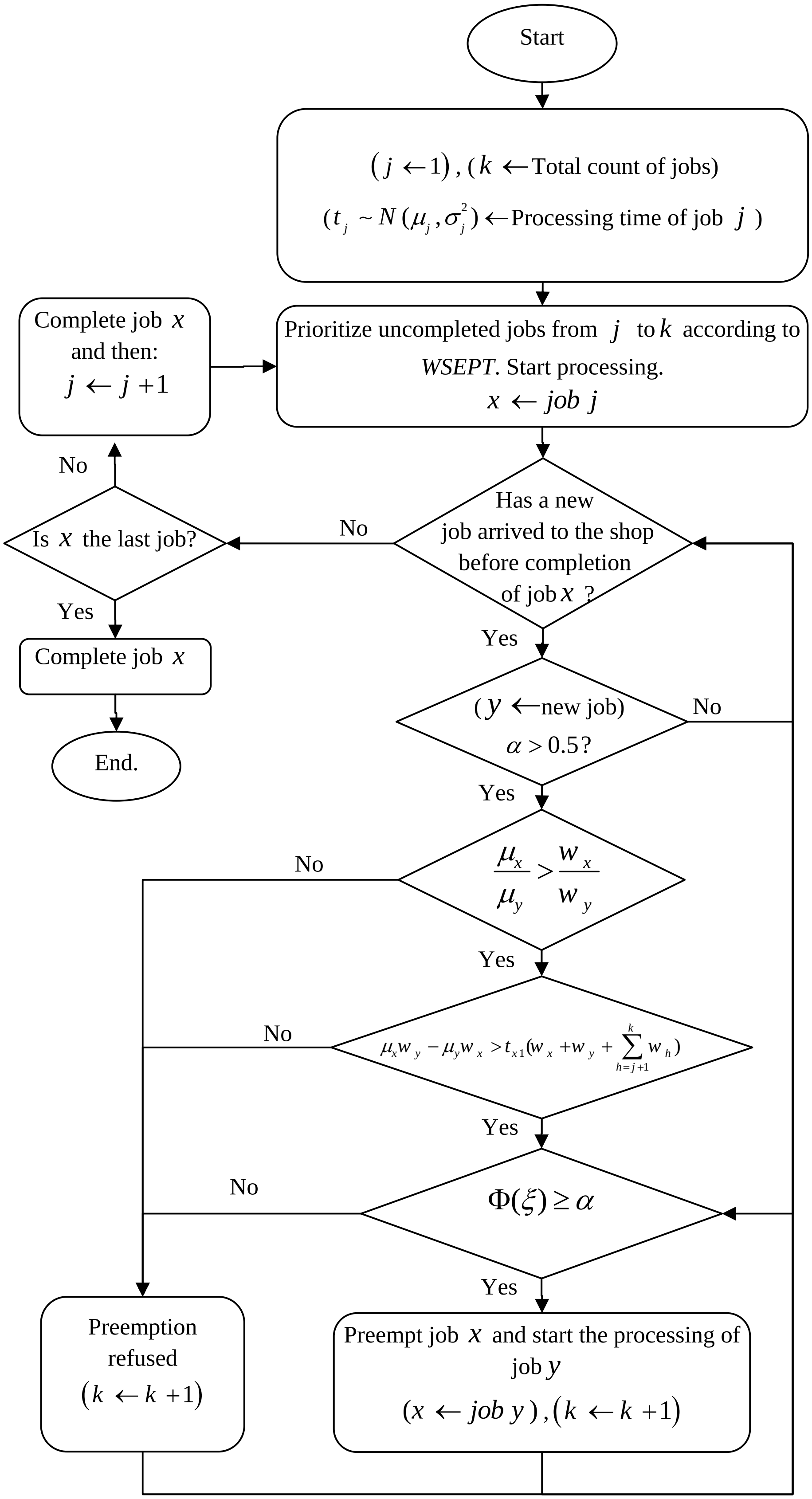
Start
$(j \leftarrow 1)$, $(k \leftarrow$ Total count of jobs)
$(t_j \sim N(\mu_j, \sigma_j^2) \leftarrow$ Processing time of job j)
Prioritize uncompleted jobs from j to k according to *WSEPT*. Start processing.
$x \leftarrow job\ j$
Complete job x and then:
$j \leftarrow j+1$
No
Has a new job arrived to the shop before completion of job x ?
No
Is x the last job?
Yes
Complete job x
End.
Yes
$(y \leftarrow$ new job)
$\alpha > 0.5$?
No
Yes
No
$\frac{\mu_x}{\mu_y} > \frac{w_x}{w_y}$
Yes
No
$\mu_x w_y - \mu_y w_x > t_{x1}(w_x + w_y + \sum_{h=j+1}^{k} w_h)$
Yes
No
$\Phi(\xi) \geq \alpha$
Yes
Preemption refused
$(k \leftarrow k+1)$
Preempt job x and start the processing of job y
$(x \leftarrow job\ y)$, $(k \leftarrow k+1)$

Figure 3. Proposed heuristic algorithm

5. Numerical example

In this section, the performance of the proposed algorithm is evaluated through numerical examples. Firstly, assume that we have 3 jobs in the shop at time zero. The processing time of job j has normal distribution with specifications given in Table 1. In this example we assume that $\alpha = 0.8$.

Table 1. Specifications of jobs, which are present in the shop at time zero

j	μ_j	σ_j	R_j	r_j	w_j	w_j/μ_j
1	6	1	7	0	5	0.83
2	15	3	14	0	3	0.2
3	12	2.3	15	0	4	0.33

In the Table 1, r_j is arrival time and R_j is the actual processing time of job j (it is not known until processing is completed). These jobs have been sorted with *WSEPT* rule and are shown in Figure 4 (4.1). In addition, 3 jobs arrive to the shop over time, which are given in Table 2.

Table 2. Jobs which arrive to the shop over time

i	μ_i	σ_i	R_i	r_i	w_i
4	3	0.5	1.5	2	8
5	5	1	2.5	9	9.5
6	9	2	5	11	1.5

The machine will start to process the jobs according to the primary sequence. Job 4 with release time $r_4 = 2$ arrives to the shop as shown in Figure 4 (4.1). Job 1 is processing at this time. Therefore,

$j = 1$

$k = 3$

$x \leftarrow job\,1$

$y \leftarrow job\,4$

$t_{x1} = 2$

First, we should check the condition given in lemma 1, in step 4 of algorithm:

$$\frac{\mu_x}{\mu_y} = 2$$

$$\frac{w_x}{w_y} = 0.625$$

$$\Rightarrow \frac{\mu_x}{\mu_y} > \frac{w_x}{w_y}$$

According the algorithm, we should go to step 5. Hence, for the inequality of $\mu_x w_y - \mu_y w_x > t_{x1}(w_x + w_y + \sum_{h=j+1}^{k} w_h)$ we have:

$\mu_x w_y - \mu_y w_x = 33$

$t_{x1}(w_x + w_y + \sum_{h=j+1}^{k} w_h) = 40$

Therefore, the preemption is refused and the priority of job 4 is determined with *WSEPT* rule between non-started jobs and the sequence would be as shown in Figure 4 (4.2). Processing of jobs is continued and during the processing of job 3, the job 5 with release time $r_5 = 9.5$ arrives to the shop as shown in Figure 4 (4.2). Thus, the parameters at this time are as follows:

$j = 3$

$k = 4$

$x \leftarrow job\,3$

$y \leftarrow job\,5$

$t_{x1} = 1$

Since inequalities $\frac{\mu_x}{\mu_y} > \frac{w_x}{w_y}$ and $\mu_x w_y - \mu_y w_x > t_{x1}(w_x + w_y + \sum_{h=j+1}^{k} w_h)$ are satisfied; the last condition needs to be examined as bellow:

$\xi = 1.87$

$\Phi(\xi) = 0.97 > 0.8$

Therefore, preemption is allowed. Job 3 is preempted and the processing of job 5 will start. Consequently, the scheduling of jobs will be as shown in Figure 4 (4.3). Job 6 arrives to the shop at time 11 and job 5 is processing at this time as shown in Figure 4 (4.3). Because the inequality $\frac{\mu_x}{\mu_y} > \frac{w_x}{w_y}$ is not satisfied, preemption is not allowed. The priority of job 6 is determined with *WSEPT* method and final schedule will be as shown in Figure 4 (4.4).

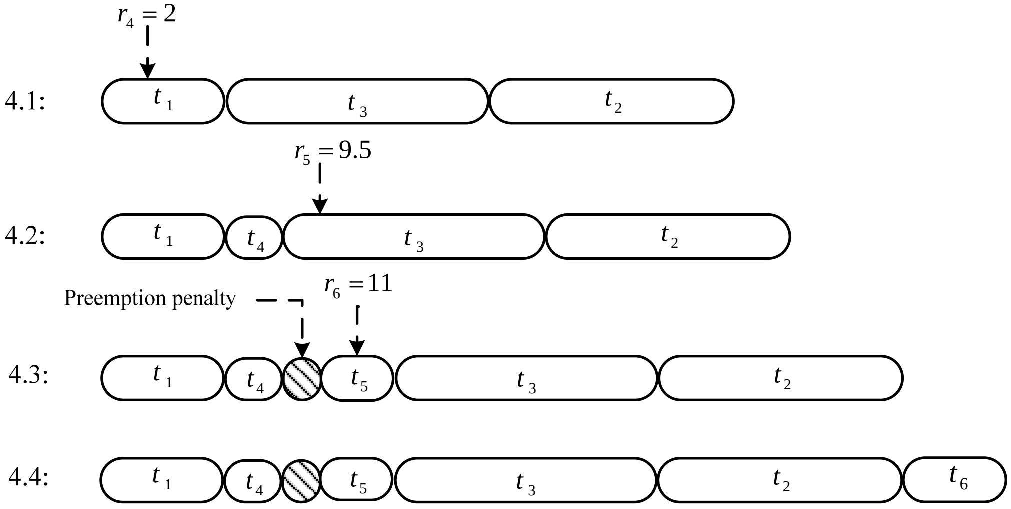

Figure 4- scheduling a sample problem with proposed algorithm

In this example, one preemption was allowed and the expected completion times of the jobs are given in Table 3.

Table 3- expected completion times of the jobs

j	μ_j	σ_j	R_j	r_j	$E(c_j)$	$r_j + E(t_j)$	w_j
1	6	1	7	0	6	6	5
4	3	0.5	1.5	2	9	5	8
5	5	1	2.5	9.5	15	14.5	9
3	12	3	15	0	27	12	4
2	15	2.3	14	0	42	15	3
6	9	2	5	11	51	20	1.5

The total weighted completion time value is computed as follows:

$$\sum w_j E(c_j^{BHM}) = \sum_{j=1}^{6} w_j (E(c_j)) = 547.5$$

$$\sum w_j E(c_j^{OPT}) = \sum_{j=1}^{6} w_j (r_j + E(t_j)) = 323.5$$

6. Performance analysis

A stochastic online method θ is a ρ*-approximation,* for some $\rho \geq 1$, if for all problem instances I , $E[\theta(\mathrm{I})] \leq \rho E[OPT(\mathrm{I})]$

Where $E[\theta(\mathrm{I})]$ and $E[OPT(\mathrm{I})]$ are the expected values that the policy θ and optimal method, respectively, achieve on instance I. The value ρ is called performance guarantee of policy θ. In this paper the trivial bound $E[OPT(\mathrm{I})] \geq \sum_j w_j [r_j + E(t_j)]$ is used for approximation factor computation and evaluation of proposed method. For performance evaluation of the proposed algorithm, we have used it on some problems with variety of sizes and the results are compared to trivial optimum (*OPT*). We have produced 10,000 problems in 100 categories with different quantities in the number of jobs, the processing time specification and the value of α coefficient. Other assumptions are as follows:

1. Release dates are generated using uniform distribution within the interval [0,3000]
2. Weights of jobs are generated using uniform distribution within the interval [1,10]
3. All jobs are preemption-restart.
4. The actual processing time is assumed to be equal to the mean of processing time.
5. We have produced 100 problems for each category and based on the comparison between results of each method, the maximum of factor ρ is calculated for these problems.

To compare the performance of the proposed method (*BHM*) against *OPT*; we first analyze the performance with respect to the number of jobs which is changed from 10 to 200 and 20 categories produced. The processing time follows a normal distribution. We assume that $\alpha = 0.8$ and the mean of processing time is a random value within the interval [40,80] and standard deviation of processing time is equal to 10. Since the mean of processing time follows uniform distribution, the expected value of the processing time is $E(\mu_j) = 60$. Table 4 and Figure 5 summarize the details of our implementations. As can be observed, the approximation factor for *BHM* is less than 2 when the number of jobs is less than 150. Note that when n extends beyond 50, the density of jobs will increase illogically and we have $n \times E(\mu_j) > 3000$ while $r_j < 3000$. Therefore, when the density of jobs increases illogically, ρ value increases faster. Moreover, in this paper ρ is calculated based on comparison of the results with trivial optimum. While calculating a lower bound for the optimal solution of the stochastic online scheduling problem with preemption penalty would result in significant improvement in the approximation factor. We have also compared the performance of the proposed method when the mean of processing time and the value of α vary. We have produced 80 categories and the number of jobs and the value of standard deviation constant.

$n = 50$

$\sigma = 10$

The results are shown in Table 5 and Figure 6. When the expected value of mean processing time, $E(\mu_j)$, increases, the proposed method provides worse solutions. When $E(\mu_j)$ exceeds 60 the density of the jobs increases illogically and the value of ρ increases faster. Also, when $0.7 \le \alpha \le 0.9$ the performance of the algorithm is better.

Table 4- the performance of the proposed algorithm versus trivial optimum

No. of jobs	$\frac{BMH}{1000}$	$\frac{OPT}{1000}$	$\rho = \frac{BMH}{OPT}$	**No. of jobs**	$\frac{BMH}{1000}$	$\frac{OPT}{1000}$	$\rho = \frac{BMH}{OPT}$
10	87.2	86.8	1.0049	110	1547.9	942.6	1.6448
20	177.4	175.6	1.0104	120	1796.1	1021.7	1.7615
30	260	255.3	1.0189	130	2108.4	1120.2	1.8858
40	352.3	340.4	1.0354	140	2391.5	1198.8	1.9985
50	454.9	426.7	1.0683	150	2781.4	1292.2	2.1567
60	577.4	517.3	1.117	160	3125.9	1377.5	2.2734
70	723.2	602.2	1.203	170	3512.4	1460.3	2.4081
80	891.8	686.1	1.302	180	3897.9	1551.1	2.5163
90	1093.3	774.2	1.4143	190	4320.3	1621.6	2.6684
100	1307.6	860.2	1.5231	200	4809.5	1730	2.7838

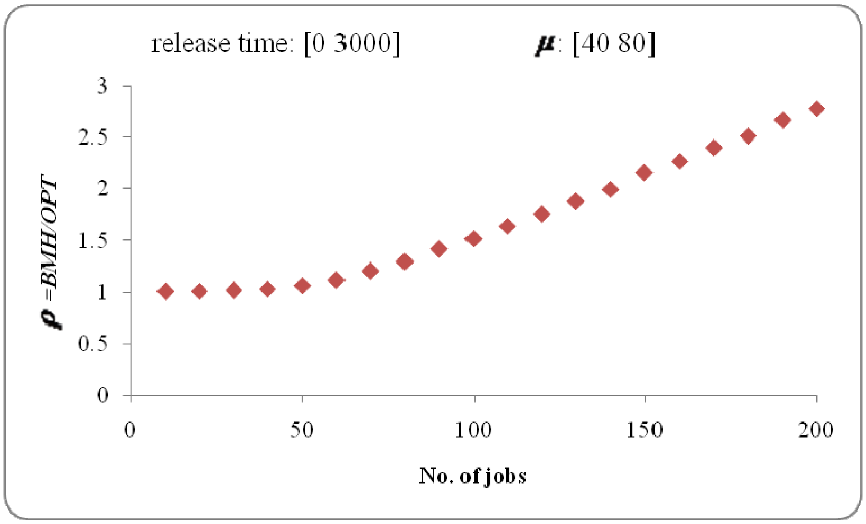

Figure 5- ρ versus the number of jobs for $\alpha = 0.8$

Table 5- The performance of *BHM* against *OPT* where the number of jobs is 50

Mean of processing time	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$
[40 45]	1.0257	1.024	1.0212	1.021	1.0209	1.0191	1.0192	1.0192
[40 50]	1.0332	1.0276	1.0258	1.0266	1.0241	1.0225	1.0231	1.0222
[40 60]	1.043	1.0398	1.036	1.0363	1.0362	1.0338	1.0345	1.0318
[40 70]	1.0608	1.0527	1.0516	1.0508	1.0494	1.0471	1.0441	1.0502
[40 80]	1.0803	1.0814	1.0734	1.0752	1.0688	1.0679	1.0653	1.0634
[40 90]	1.101	1.1003	1.0933	1.0983	1.0956	1.0902	1.0952	1.0907
[40 100]	1.1409	1.1324	1.1315	1.1221	1.1225	1.1205	1.113	1.1128
[40 200]	1.4954	1.5108	1.5082	1.5119	1.4656	1.4812	1.4934	1.4979
[40 500]	2.6798	2.6097	2.599	2.6173	2.6067	2.5883	2.6394	2.6131
[40 1000]	4.2528	4.2194	4.1734	4.1968	4.1599	4.1503	4.1011	4.1088

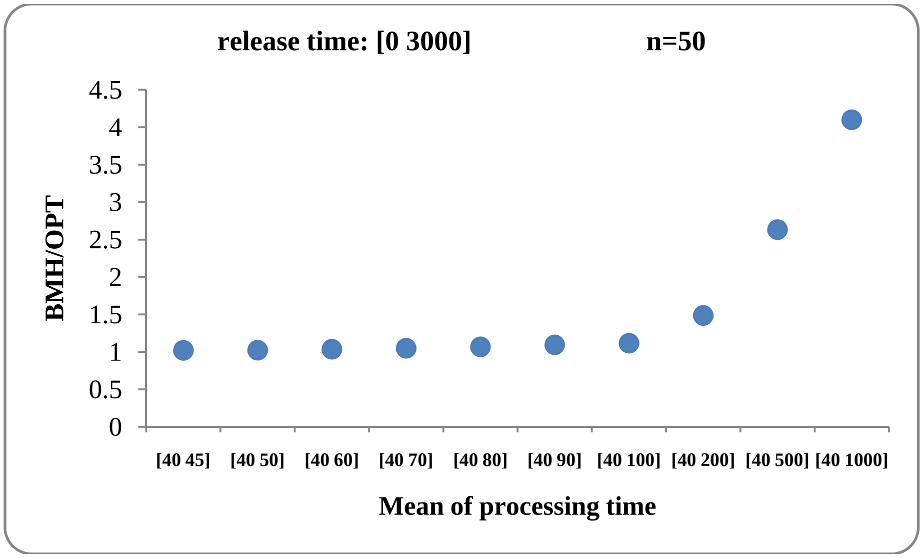

Figure 6- ρ versus mean of processing time for $\alpha = 0.7$

7. Conclusion

In this paper, the stochastic online scheduling problem with preemption penalty, with objective function of minimizing the expected value of the total weighted completion time on single machine had been investigated. The problem has specifications such as being online, preemption penalty and stochastic processing times. Moreover, a heuristic algorithm was presented and the implementation of the proposed method was demonstrated using some numerical examples. The performance of the proposed method in this paper had been compared to trivial optimum and the approximation factor had been computed. Primary results indicated that the proposed method had the approximation factor less than 2 for a wide range of problems. The proposed method utilized the properties of the normal distributions and this method can be used as a heuristic method for other distributions, as long as their means and variances are available. Scheduling with preemption penalties is a novel research area in scheduling field especially for online problems and we strongly believe that this research could be well extended for problems with two uniform machines or flow shop problem and problems with stochastic setup times.

8. References

[1] C.N. Potts, L.N.V. Wassenhove, J. Oper. Res. Soc. 43,pp-395(**1992**).

[2] Z. Liu, T.C.E. Cheng, Information Process. Lett., 82, pp-107(**2002**).

[3] F. Zheng, W. Dai, P. Xiao, Y. Zhao,. Competitive strategies for on-line production order disposal problem. In: Proc. of 1st International Conference on Algorithmic Applications in Management., pp 46(**2005**).

[4] F. Zheng, Y. Xu, E. Zhang, J Comb Optim 13,pp-189(**2007**).

[5] P.Y.F. Stanley, Inf Process Lett 108, pp-214(**2008**).

[6] M. Heydari, S.J. Sadjadi, E. Mohammadi, Minimizing Int J Adv Manuf Technol, 47, pp-227(**2009**).

[7] D. Chazan, A.G. Konheim, B. Weiss. Journal of Combinatorial Theory, 5, pp-344(**1968**).

[8] A.G. Konheim. Probability Theory and Related Fields, 9, pp-112(**1968**).

[9] K.C. Sevcik. Journal of the ACM, 21, pp-65(**1974**).

[10] G. Weiss. Advances in Applied Probability, 27, pp-827(**1995**).

[11] T. Kämpke. Operations Research 37 (1), pp-126(**1989**).

[12] N. Megow, T. Vredeveld. Approximation results for preemptive stochastic online scheduling, Technical Report 8, Technische Universität Berlin(**2006**).

[13] Manzhan Gu , Xiwen Lu. Information Processing Letters 109, pp-369(**2009**).

[14] J. Labetoulle, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinooy Kan. Preemptive scheduling of uniform machines subject to release dates. In W. R. Pulleyblank, editor, Progress in Combinatorial Optimization, pages 245–261. Academic Press, New York(**1984**).

[15] J. K. Lenstra, A. H. G. Rinooy Kan, and P. Brucker. Annals of Discrete Mathematics, 1,pp-243(**1977**).

[16] N. Megow. Coping with incomplete information in scheduling stochastic and online models, Ph.D. Thesis, Technische Universität Berlin, Published by Cuvillier Verlag G**ö**ttingen, Germany(**2007**).

[17] Heydari M. Mohammadi E. Single machine scheduling with fuzzy preemption penalties, The Journal of Mathematics and Computer Science Vol .2 No.1, pp-122 (**2011**).