Scheduling with uncertain durations: generating β-robust schedules using constraint programming

Christine Wei Wu and **Kenneth N. Brown**
Cork Constraint Computation Center,
Department of Computer Science,
University College Cork, Ireland
{cww1, k.brown}@cs.ucc.ie

J. Christopher Beck
Toronto Intelligent Decision Engineering
Laboratory, Dept. of Mechanical and
Industrial Engineering,University of Toronto,
Canada. jcb@mie.utoronto.ca

Abstract

Many real-world scheduling problems are subject to change, and scheduling solutions should be robust to those changes. We consider a single-machine scheduling problem where the processing time of each activity is characterized by a normally-distributed random variable, and we attempt to minimize flowtime. We develop an initial constraint model for generating the β-robust schedule - the schedule that has highest probability of producing a flowtime less than a stated bound. Experiments with this initial model show that a constraint-based approach is feasible, but that better propagation methods will be required.

Introduction

Most scheduling research considers problems that are static and certain – all the activities and their durations are known in advance and do not change as the solution is being executed. However, many real-world scheduling problems are subject to change: new jobs arrive, resources fail, or tasks take longer than expected. If these changes are significant, then optimal solutions to the original problem may turn out to be poor in practice. For this reason, it may be better to generate solutions that are robust to the likely changes. A β-*robust* schedule (Daniels & Carrillo 1997) is one that has maximum probability of achieving a given performance level (e.g. total flowtime less than a threshold). Alternatively, we may want to find the best performance that a solution will deliver with a given confidence level.

Constraint-based methods have proven to be very effective in a wide range of industrial scheduling problems. The advantage comes from the flexibility of the modeling language, and the ability of the solvers to deliver effective performance despite the presence of a wide range of different constraints and objectives. Again, though, most constraint based research assumes static and certain problems. In this paper, we consider how to model β-robustness in a constraint modelling language, and we show how to search for β-robust schedules.

In particular, we consider single machine problems, where the processing time of each task is uncertain, but can be characterized by a normally-distributed random variable. We consider flowtime (the amount of time the tasks remain in the system) as the main criterion. The simplest approach would focus on optimizing the expected total flowtime. However, this ignores the variance in the task durations, which may be significant. For any given schedule, we will measure the probability of the total flowtime being less than a target level. We will then generate (i) a schedule which maximizes the probability, or (ii) a schedule which optimizes the target level that can be achieved with a given probability.

The paper is structured as follows: first, we briefly review techniques for scheduling under uncertainty; we then consider flowtime as a performance measure for schedules with uncertain task durations; we give a formal definition of β-robustness; we present our initial constraint models for the β-robust scheduling problem; and finally we report on some experiments with the model.

Background

A number of approaches have been proposed to handle uncertain scheduling problems. Redundancy-based Scheduling generates schedules with temporal slack so that unexpected events during execution can be handled by using that reserved slack (Davenport, Gefflot, & Beck 2001; Gao 1995). Contingent scheduling anticipates likely disruptive events and generates multiple schedules which optimally respond to the anticipated events (Drummond, Bresina, & Swanson 1994; Fowler & Brown 2003). Probabilistic scheduling uses probabilities over possible events, and searches for schedules which optimize the expected value of some performance measure (Daniels & Carrillo 1997; Walsh 2002; Beck & Wilson 2004; 2005). A number of approaches use sampling and scenarios, in order to produce robust decisions (Bent & Hentenryck 2004; Beck & Wilson 2004).

In particular, Daniels and Carrillo (Daniels & Carrillo 1997) introduced the concept of the β-robust schedule for a single machine scheduling problem with processing time uncertainty. They solved the problem by a branch-and-bound method with dominance rules, and heuristics for branch selection. The total flowtime was used to measure the performance of solutions, which will be explained in the next section.

The flowtime of a schedule

In a single machine scheduling problem, in which each job consists of a single task, a machine can only process one job at a time, and a job cannot be interrupted once started, a solution is a sequence of the jobs, and we assume the jobs are executed in sequence with no delay between them. Suppose we have a sequence $J_1, J_2, \ldots J_n$. Each job J_i has an arrival time A_i (its earliest possible start time), a start time ST_i, a duration d_i, and an end time E_i. We assume that each job is available for processing at time 0 (i.e. $A_i = 0$).

We note the following simple relations: $E_i = ST_i + d_i$, $ST_1 = 0$, $ST_i = E_{i-1}$, and hence $E_i = \sum_{j=1}^{i} d_j$.

The *flowtime* is the total time the jobs are in the system:

$$TFT = \sum_{i=1}^{n}(E_i - A_i)$$

Because we assume $Ai = 0$, we can rewrite the equation for total flowtime as follows:

$$\begin{aligned} TFT &= \sum_{i=1}^{n} E_i && (1) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{i} d_j && (2) \\ &= \sum_{i=1}^{n}(n+1-i) * d_i && (3) \end{aligned}$$

We now assume that each job J_i's duration is an independent normally distributed random variable $d_i \sim N(\mu_i, \sigma_i^2)$. We assume that the jobs will still be executed in the given sequence, regardless of the actual values of the durations.

We note that for any two independent random variables $X \sim N(\mu_x, \sigma_x^2)$ and $Y \sim N(\mu_y, \sigma_y^2)$, and two constants a and b, the sum $aX + bY$ is also a normally distributed random variable, such that $aX+bY \sim N(a\mu_x+b\mu_y, a^2\sigma_x^2 + b^2\sigma_y^2)$.

Since the activity durations are independent normally distributed random variables, and flowtime is a linear combination of durations, then for any particular sequence of jobs, the flowtime is also a normal random variable. From (3):

$$TFT \sim N(\sum_{i=1}^{n}(n-i+1)\mu_i, \sum_{i=1}^{n}(n-i+1)^2\sigma_i^2)$$

β-robust schedules

For scheduling problems with uncertainty, we must decide the criteria by which the scheduling solutions will be judged. The simplest criterion is the expected flowtime (or the average actual flowtime over a number of runs). In this case, the scheduler only needs to consider the expected parameters of the individual jobs. However, in real settings, some form of service level may be more important – what level of confidence can a customer or manager have in predicted performance levels? Rather than gambling on the expected performance, it may be more useful to give a lower limit on the performance, and to state the confidence in being able to achieve that level. In this case, it is not enough to know the expected values of the job parameters – the scheduler must also reason about the variance of those parameters in order to determine the variance of the schedule as a whole.

For example, consider the simple problem consisting of three jobs $\{x, y, z\}$, with uncertain durations $\{d_x \sim N(9, 2), d_y \sim N(5, 1), d_z \sim N(8, 7)\}$. The sequence $s_e = \langle y, z, x\rangle$ has a flowtime which is distributed as $N(40, 39)$. 40 is, in fact, the smallest expected flowtime possible for this problem. An alternative sequence, $s_\beta = \langle y, x, z\rangle$, has flowtime $\sim N(41, 24)$, and thus has a higher expected flowtime. However, suppose we now introduce a desired maximum flowtime of (for example) 51: the scheduler will incur a penalty if the actual schedule has a flow time greater than 51. Sequence s_e has a probability of 0.04 of producing a flowtime greater than 51, while s_β has a probability of just 0.02 of delivering a flowtime greater than 51, and thus s_β is likely to be less expensive. s_β is the *β-robust* (Daniels & Carrillo 1997) schedule for the maximum flowtime of 51 - that is, it has the highest probability of delivering a flowtime no greater than 51. In addition, for the confidence level of 0.98, s_β also delivers the minimal flowtime limit (51).

Definition 1. *For the single machine scheduling problem with n jobs, with normally-distributed uncertain durations, and with a flowtime limit S, the β-robust scheduling problem is to find the sequence, s, which maximizes the probability of the flowtime being less than S. That is, find the s that maximizes $Prob(flowtime(s) \le S)$ (Daniels & Carrillo 1997).*

First, we show how to compute $Prob(\,flowtime \le S\,)$ for an arbitrary sequence of the n jobs. Since the random variables in the problem are normally distributed, we can use the formula below to compute the probability of $flowtime \le S$, where μ is the mean flowtime of the sequence, and σ^2 is its variance:

$$\phi(x \le X) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{X} e^{\frac{-(x-\mu)^2}{2\sigma^2}}dx$$

An arbitrary normal distribution can be converted to a standard normal distribution by changing variables to $z = (x - \mu)/\sigma$, so the normal distribution function becomes:

$$\begin{aligned} \phi(x \le X) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{\frac{-t^2}{2}}dt \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{0} e^{\frac{-t^2}{2}}dt + \phi(z) \\ &= \frac{1}{2} + \phi(z) \end{aligned}$$

where

$$\phi(z) = \frac{1}{\sqrt{2\pi}}\int_{0}^{z} e^{\frac{-t^2}{2}}dt.$$

Hence, the probability of $flowtime \le S$ can be computed by

$$Prob(\,flowtime \le S\,) = \frac{1}{2} + \phi(z) \qquad (4)$$

where $S \geq 0$ and

$$z = \frac{S - mean(flowtime)}{\sqrt{var(flowtime)}}. \quad (5)$$

For each possible schedule, we can compute the mean and variance of the flowtime by $mean(flowtime) = \sum_{i=1}^{n}(n-i+1)\mu_i$ and $var(flowtime) = \sum_{i=1}^{n}(n-i+1)^2\sigma_i^2$ as in equation (3). Then, $\phi(z)$ can be obtained by checking z in the standard normal distribution table (Z-table).

Alternatively, there is a simple approximation of $\phi(z)$ which is good to two decimal places (Weisstein 2006), given by

$$\phi(z) \approx \varphi(z) \begin{cases} 0.1z(4.4-z) & (0 \leq z \leq 2.2) \\ 0.49 & (2.2 < z < 2.6) \\ 0.50 & (z \geq 2.6) \end{cases} \quad (6)$$

Theorem 1. *$\varphi(z)$ increases on $[0, +\infty)$. For proof, see appendix.*

The β-robust schedule is one of those alternative sequences of the jobs, such that it has the maximum probability of $flowtime \leq S$. To find a β-robust schedule, we need to have an objective function to maximize the probability. We use the approximation of $\phi(z)$ to compute the probability, because it simplifies the calculation. If $\phi(z)$ is increasing on $[0, +\infty)$, maximizing the probability of $flowtime \leq S$ is the same as maximizing z.

$$\begin{aligned} objective &= max(\ probability\ (flowtime \leq S)\ ) \\ &= max(\ \frac{1}{2} + \phi(z)\ ) \\ &= \frac{1}{2} + max(\ \phi(z)\ ) \\ &\approx \frac{1}{2} + max(\ \varphi(z)\ ) \\ &= \frac{1}{2} + \varphi(\ max(z)\ ). \end{aligned}$$

With above analysis and calculations, we are ready to introduce our constraint models for the β-robust scheduling problem.

Constraint models

We first consider the original β-robust scheduling problem described in (Daniels & Carrillo 1997) as a CSP, and then we propose a variable ordering heuristic.

The Constraint Model is shown in Figure 1. We assume a set $\{J_1, J_2, \ldots J_n\}$ of jobs, each with a normally-distributed random variable duration $D_i \sim N(\mu_i, \sigma_i^2)$. Differing from the previous sections, we now do not assume that the jobs are scheduled in the given sequence. With each job J_i, we associate a position variable, Pos_i, with domain $\{1, 2, \ldots, n\}$. The position variable Pos_i represents the position of J_i in the sequence: for instance, $Pos_2 = 3$ states that J_2 is scheduled to be the third job to start on the machine. Besides position variables, we also introduce additional variables for computing flowtime mean and variance and then the probability. For any job i, $meanFT_i$ is the mean flowtime from the first job up to J_i. The value

Figure 1: the Constraint Model.

Variables:

Job positions: $Pos_1, ..., Pos_n$
Job mean flowtimes: $meanFT_1, ..., meanFT_n$
Job mean flowtime contributions:
$meanFTContrib_1, ..., meanFTContrib_n$
Job variance flowtime contributions:
$varFTContrib_1, ..., varFTContrib_n$

Constraints:

`allDifferent`(Job positions)

let $meanFT_0 = 0$,
$Pos_j > Pos_i \Rightarrow meanFT_i \leq meanFT_j - \mu_j$
$Pos_j = Pos_i + 1 \Rightarrow meanFT_j = meanFT_i + \mu_j$
$meanFTContrib_i = (n - Pos_i + 1)\mu_i$
$varFTContrib_i = (n - Pos_i + 1)^2\sigma_j^2$
$mean(flowtime) = \sum_{i=1}^{n} meanFT_i$
$mean(flowtime) = \sum_{i=1}^{n} meanFTContrib_i$
$var(flowtime) = \sum_{i=1}^{n} varFTContrib_i$

Dominance constraints:
for $0 < i, j \leq n$, if $Pos_i < Pos_j$
$\mu_i > \mu_j \Rightarrow \sigma_i^2 < \sigma_j^2$
$\sigma_i^2 > \sigma_j^2 \Rightarrow \mu_i < \mu_j$.

$objective = max(z) = max(\ \frac{S - mean(flowtime)}{\sqrt{var(flowtime)}}\ )$

of $meanFT_i$ is an integer in the interval $[\mu_i, totalMean]$, where $totalMean = \sum_{i=1}^{n} \mu_i$. The formula (3) indicates that flow time can be also viewed as the sum of the contributions from all jobs. We define flowtime contribution of J_i as $FTContrib_i = (n - Pos_i + 1)D_i$. $meanFTContrib_i$ and $varFTContrib_i$ are the mean and variance of the flowtime contributions from J_i. The former has an integer value in $[\mu_i, n\mu_i]$, and the latter has a value in $[\sigma_i^2, n^2\sigma_i^2]$. The goal is to sequence those jobs, i.e. assign a distinct value to each Pos_i, such that the likelihood of the sequence (schedule) to achieve a fixed system performance level S is optimized, i.e. $Max(\ Probability(\ X \leq S\ )\ )$, where X is flowtime. Alternatively, we can look for a schedule with the optimal system performance S for a fixed probability, i.e. $Min(\ S\ )$ such that $Probability(\ X \leq S\ ) \geq C$, where C is the fixed probability.

We show how to achieve the first goal, i.e. optimizing the probability for a fixed performance. Firstly, we have a permutation constraint that ensures each job takes a different position in the sequence. This can be implemented as a global all-different constraint on all the Pos_i. Also, we add further constraints during search: If job i is located before job j in the sequence (i.e. $Pos_i < Pos_j$), we have

$$ST_i + D_i \leq ST_j,$$

where ST_i and ST_j are start times of i and j respectively, and so

$$FT_i \leq FT_j - D_j$$

where FT_i is the flowtime from the first job up to job i. Since $D_i \sim N(\mu_i, \sigma_i^2)$ and $D_j \sim N(\mu_j, \sigma_j^2)$ are independent random variables, we get

$$meanFT_i \leq meanFT_j - \mu_j.$$

Also if we consider the flowtime as a sum of contributions from each job, from formula (3), we have

$$\begin{aligned} mean(flowtime) &= \sum_{i=1}^{n} meanFTContrib_i \\ &= \sum_{i=1}^{n} (n - Pos_i + 1)\mu_i, \\ var(flowtime) &= \sum_{i=1}^{n} varFTContrib_i \\ &= \sum_{i=1}^{n} (n - Pos_i + 1)^2 \sigma_i^2. \end{aligned}$$

Note that for each job, the flowtime from the first up to the job, FT_i, is different from its flowtime contribution $FTContrib_i$. For example, let $meanFT_0 = 0$, for $0 < i, j \leq n$,

$$Pos_j = Pos_i + 1 \Rightarrow meanFT_j = meanFT_i + \mu_j,$$

and for $0 < i \leq n$,

$$meanFTContrib_i = (n - Pos_i + 1)\mu_i.$$

However, the sum of $meanFT_i$ is still equal to the mean of total flowtime, i.e. $mean(flowtime) = \sum_{i=1}^{n} meanFT_i = \sum_{i=1}^{n} meanFTContrib_i$. With those additional variables, we can use formula (4), (5) and (6) to compute the probability of a schedule's actual flowtime being less than S.

We are also able to impose some dominance constraints as in figure 1, using the properties of the β-robust schedule.

Theorem 2. *In a β-robust schedule, if job i with $D_i \sim N(\mu_i, \sigma_i^2)$ precedes job j with $D_j \sim N(\mu_j, \sigma_j^2)$, then either the mean duration of job i, μ_i, is no greater than the mean duration of job j, μ_j, or the duration variance of job i, σ_i^2, is no greater than the duration variance of job j, σ_j^2, that is $\mu_i \leq \mu_j$ or $\sigma_i^2 \leq \sigma_j^2$. (see Appendix for the proof)*

With this property, we post further constraints during search: if job i starts before job j (i.e. $Pos_i < Pos_j$), then

$$\mu_i > \mu_j \Rightarrow \sigma_i^2 < \sigma_j^2, \qquad \sigma_i^2 > \sigma_j^2 \Rightarrow \mu_i < \mu_j.$$

So far, we have presented our constraint model to achieve the first objective. Our second objective is to minimize system performance S such that there exists a schedule that can achieve S with a fixed probability. That is $Min(\,S\,)$ such that $Probability(\,X \leq S\,) \geq C$, where C is the fixed probability. Using the same constraint model, we can get z value from formula (4) and (6)

$$z = \varphi^{-1}(C - \frac{1}{2}).$$

Then, from formula (5), we have a new objective function

$$min(S) = min(\,z * \sqrt{var(flowtime)} + mean(flowtime)\,).$$

Besides those constraints we discussed above, we also implement a variable ordering heuristic to guide search. From formula (5), we can see that the β-robust schedule has the optimized combination of $mean(flowtime)$ and $var(flowtime)$. In order to find the β-robust schedule more quickly, we prefer to first schedule a job i, which has shorter mean processing time μ_i and smaller variance σ_i^2. We use a family of variable ordering heuristics, ordering the jobs by increasing $\mu_i + q * \sigma_i^2$, selecting a value for q based on the problem characteristics. For the first objective (maximizing the probability), we start by finding the SEPT (shortest expected processing time) schedule; we then compute the probability P of it having a flowtime less than S; and from P we select a value for q from a lookup table based on previous experiments with other problems. For the second objective (minimizing the flowtime target achievable by a given probability), we base q on the probability. In both cases, for higher probabilities, we expect the variance to be more significant, and so we choose higher values of q which give increasing weight to the duration variance in the variable ordering. Example values for q are 0.3,0.6, and 1.0 for probabilities of 0.85, 0.95 and 0.99. Note that this variable ordering heuristic does not improve the total solving speed (i.e. the time of finding the schedule and proving it is the optimal), but does shorten the time to find the optimal solution.

Discussion and Experimental results

We implemented the β-robust scheduling problem as a constraint satisfaction problem using ILOG Scheduler and Solver 6.0. Our first aim is to verify our initial constraint model, and so we expect to see the same pattern of results as obtained by (Daniels & Carrillo 1997). Secondly, we want to determine whether or not a constraint model is feasible for such problems, and so we hope to see runtimes of a similar order of magnitude. If we succeed in both aims, we will then investigate more sophisticated constraint models.

We consider problems with either 10 or 15 jobs, using the same experimental setup as (Daniels & Carrillo 1997). The mean processing time for each job i is randomly drawn from a uniform distribution of integers on the interval $\mu_i \in [10, 50\delta_1]$; the processing time variance of job i is then randomly drawn from a integer interval $\sigma_i^2 \in [0, \frac{1}{9}\mu_i^2\delta_2]$. The parameter δ_1 and δ_2 control the variability in the average processing times in the test problems, which can both take any value of 0.4, 0.7 or 1.0. We are interested in schedules that yield acceptable performance with probability approximately 0.85, 0.95 or 0.99. Ten instances are generated randomly for each combination of number of jobs, δ_1, δ_2, and probability level, resulting in a total 540 test problems.

Table 1 contains the results for our constraint methods and the corresponding figures taking directly from Daniels and Carrillo (Daniels & Carrillo 1997). The CPU is the computation time for finding and proving the β-robust schedule. Since Daniels and Carrillo performed the experiment

Table 1: Computational performance of β-robust solution procedure.

		Constraint model			Branch-and-bound		
total jobs	prob. level	CPU (sec.)	Avg. abv. SEPT (%)	Max. abv. SEPT (%)	CPU (sec.)	Avg. abv. SEPT (%)	Max. abv. SEPT (%)
10	0.85	0.1	0.1	0.4	0.2	0.1	0.8
	0.95	0.1	0.3	1.7	0.2	0.3	1.9
	0.99	0.1	0.5	1.9	0.3	0.6	2.5
15	0.85	6.6	0.1	0.3	1.0	0.1	0.5
	0.95	5.3	0.2	0.7	1.7	0.2	1.0
	0.99	14.3	0.3	1.5	2.1	0.4	1.9

Table 2: The comparison of finding the optimal result and proving that is optimal.

		Time(s)		Time(%)	
total jobs	prob. level	search	prove	search	prove
15	0.85	0.26	6.94	5.10	94.90
	0.95	0.18	6.62	5.38	94.62
	0.99	0.93	12.61	11.84	88.16

on a 486 personal computer, our CPU time is not directly comparable with theirs but should give us an indication of whether or not a constraint-based approach is feasible. Table 1 also shows the differences (in average and in maximum deviation) between the mean processing time of the β-robust schedule and the shortest expected processing time (SEPT).

The results in table 1 show that we do have a similar pattern in term of the mean flowtime of the β-robust schedule compared to the SEPT schedule. In addition, our CPU time is comparable for the smaller problems, but is poorer for the larger problems. This indicates that a constraint-based approach may be feasible, but that a more sophisticated model with better propagation will be required. We set up a further experiment to determine the effort require to prove that the solution is optimal. For each probability level, we experimented with 90 problems generated the same way as table 1. We also used the variable ordering heuristic as described in constraint models section. The time of finding the optimal solution and the time to prove it is optimal were recorded in each case. Table 2 shows that it takes little time to find the best solution but usually a long time to prove if it is the β-robust schedule. We believe that a problem is hard for our model if it has many jobs with similar duration mean and variance. The program is able to do little propagation, and thus spends a lot of time trying different permutations of the jobs for no benefit.

With the general model, we can also give the minimum system performance S for a problem, so that the jobs in the problem can be scheduled to achieve the minimized S with

Table 3: Computational performance of finding the minimum system performance of required probability level (15 jobs).

		Constraint model		
number of jobs	probability level	CPU (sec.)	Avg. abv. SEPT(%)	Max. abv. SEPT(%)
15	0.85	3.01	0.04	0.24
	0.95	4.39	0.16	0.87
	0.99	7.51	0.28	1.11

a desired probability level. Table 3 shows the time to find the minimum S for 15 jobs problems and the corresponding mean flowtime over the SEPT.

Daniels and Carrillo compared the mean flowtime of the β-robust schedule and the optimal expected flowtime. That indicates how much worse the β-robust schedule is compared to the SEPT schedule. It seems more reasonable to compare the probabilities of achieving the system performance of the β-robust schedule with the SEPT schedule. That is

$$\begin{aligned} &[probability(\,flowtime_{\beta} \le S\,) \\ - \;& probability(\,flowtime_{SEPT} \le S\,)] \\ \div \;& probability(\,flowtime_{SEPT} \le S\,) \end{aligned}$$

which indicates the benefit of using the β-robust schedule. In fact, when some jobs have the same mean, there are multiple possible schedules with shortest expected processing time, but each will have different robustness levels. Daniels and Carrillo did not report the detail of how they generated the SEPT schedule. In our experiments, we used the existing variable ordering, breaking ties lexicographically.

Future work

We are currently working on the *dual* model of the original (primal) model, and a third model which channels between the other two (Hnich, Smith, & Walsh 2004). We believe using the combined model will help us to improve the solving speed. We also plan to investigate better bounds for pruning branches at the top of the search tree, better heuristics to guide the search, and the construction of a global constraint for achieving β-robustness. We are also conducting an investigation into the characteristics of the problems which make some of them much harder to solve than others. Finally, we plan to extend this work to consider problems with multiple machines and with non-zero arrival times, for which the probability calculations reported here will not apply.

Conclusion

In this paper, we presented a general constraint model for the β-robust scheduling problem, which allows us to produce schedules which are robust to uncertainty in the durations of tasks. With flowtime as the performance measure, we can optimize the probability and find a most promising schedule to satisfy the system performance requirement; or we can optimize the performance level for a fixed probability. Our initial model demonstrates that a constraint-based approach is feasible for this problem, but that more more sophisticated models are required for good performance.

Appendix

Theorem 1. *$\varphi(z)$ increases on $[0, +\infty)$.*

Proof. For all $b > a > 2.2$, it is trivial to see $\varphi(b) \geq \varphi(a)$. For all $0 \leq a < b \leq 2.2$,

$$\begin{aligned}\varphi(b) - \varphi(a) &= 0.1\, b(\, 4.4 - b\,) - 0.1\, a(\, 4.4 - a\,) \\ &= 0.1(\, 4.4b - b^2 - 4.4a + a^2\,) \\ &= 0.1[\, 4.4(\, b - a\,) + (\, a + b\,)(\, a - b\,)\,] \\ &= 0.1(\, b - a\,)(\, 4.4 - a - b\,).\end{aligned}$$

Since $b > a$,

$$\Rightarrow (b - a) > 0$$

and $0 \leq a, b \leq 2.2$

$$\Rightarrow 0 \leq a + b \leq 4.4$$

$$\Rightarrow 4.4 - a - b \geq 0.$$

Hence,

$$\varphi(b) - \varphi(a) \geq 0$$

for all $0 \leq a < b \leq 2.2$, i.e. $\varphi(z)$ increases on $[0, +\infty)$. □

Theorem 2. *In a β-robust schedule, if job i with $D_i \sim N(\mu_i, \sigma_i^2)$ precedes job j with $D_j \sim N(\mu_j, \sigma_j^2)$, then either the mean duration of job i, μ_i, is no greater than the mean duration of job j, μ_j, or the duration variance of job i, σ_i^2, is no greater than the duration variance of job j, σ_j^2, that is $\mu_i \leq \mu_j$ or $\sigma_i^2 \leq \sigma_j^2$.*

Proof. (by contradiction)
Assume X is a β-robust schedule of n jobs, and there exist two jobs i and j in X such that i precedes j, $\mu_i > \mu_j$ and $\sigma_i^2 > \sigma_j^2$.

Suppose i's position in X is a ($1 \leq a \leq n$) and j's position is b ($1 \leq b \leq n$).
Since i precedes j, we have:

$$1 \leq a < b \leq n.$$

Also, the expected flowtime contribution from i, $meanFTContrib_i$, is $(n - a + 1)\mu_i$ and the expected flowtime contribution from j, $meanFTContrib_j$, is $(n - b + 1)\mu_j$. Now, we consider only swap i and j but keep other jobs' positions unchanged in X to get schedule X'. In X', i's position is b and j's position is a. Then,

$$meanFTContrib'_i = (n - b + 1)\mu_i$$

and

$$meanFTContrib'_j = (n - a + 1)\mu_j.$$

The difference between X and X' in term of the expected total flowtime is:

$$\begin{aligned} & mean(flowtime) - mean(flowtime)' \\ = \; & meanFTContrib_i + meanFTContrib_j \\ & + meanFTContrib_{rest} - (\, meanFTContrib'_i \\ & + meanFTContrib'_j + meanFTContrib_{rest}\,) \\ = \; & meanFTContrib_i + meanFTContrib_j \\ & - meanFTContrib'_i - meanFTContrib'_j \\ = \; & (n - a + 1)\mu_i + (n - b + 1)\mu_j \\ & - (n - b + 1)\mu_i - (n - a + 1)\mu_j \\ = \; & (n - a + 1)(\mu_i - \mu_j) + (n - b + 1)(\mu_j - \mu_i) \\ = \; & (b - a)(\mu_i - \mu_j)\end{aligned}$$

where $meanFTContrib_{rest}$ is the expected flowtime contributions from the jobs other than i and j, which is the same in both X and X'.
Because in assumption $\mu_i > \mu_j$ and $b > a$, $(b - a)(\mu_i - \mu_j) > 0$, that is:

$$mean(flowtime) > mean(flowtime)'.$$

Similarly, in schedule X, the variance of flowtime contribution from i, $varFTContrib_i$ is $(n - a + 1)^2\sigma_i^2$; the variance of flowtime contribution from j, $varFTContrib_j$, is $(n - b + 1)^2\sigma_j^2$.
For schedule X', $varFTContrib'_i = (n - b + 1)^2\sigma_i^2$ and $varFTContrib'_j = (n - a + 1)^2\sigma_j^2$. Then,

$$\begin{aligned} & var(flowtime) - var(flowtime)' \\ = \; & varFTContrib_i + varFTContrib_j \\ & - varFTContrib'_i - varFTContrib'_j \\ = \; & (n - a + 1)^2\sigma_i^2 + (n - b + 1)^2\sigma_j^2 \\ & - (n - b + 1)^2\sigma_i^2 - (n - a + 1)^2\sigma_j^2 \\ = \; & (2n + 2 - a - b)(b - a)(\sigma_i^2 - \sigma_j^2)\end{aligned}$$

Since $\sigma_i^2 > \sigma_j^2$, $b > a$ and $2 \leq a + b \leq 2n$, we get:

$$var(flowtime) > var(flowtime)'.$$

Hence the z value of X is

$$z = \frac{S - mean(flowtime)}{\sqrt{var(flowtime)}}$$

and for X' that is

$$z' = \frac{S - mean(flowtime)'}{\sqrt{var(flowtime)'}}.$$

Clearly, $z < z'$, which means schedule X' has a higher probability to achieve the fixed system performance S than schedule X. That contradicts to X is a β-robust schedule for the n jobs.
Therefore the statement has been proved. □

References

Beck, J. C., and Wilson, N. 2004. Job shop scheduling with probabilistic durations. *Proceedings of the Sixteenth European Conference on Artificial Intelligence* 652–656.

Beck, J. C., and Wilson, N. 2005. Proactive algorithms for scheduling with probabilistic durations. *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence* 1201–1206.

Bent, R., and Hentenryck, P. V. 2004. Online stochastic and robust optimization. *Ninth Asian Computing Science Conference* 286–300.

Daniels, R. L., and Carrillo, J. E. 1997. Beta-robust scheduling for single-machine systems with uncertain processing times. *IIE Transactions* 29:977–985.

Davenport, A. J.; Gefflot, C.; and Beck, J. C. 2001. Slack-based techniques for robust schedules. *Proceedings of the Sixth European Conference on Planning* 7–18.

Drummond, M.; Bresina, J. L.; and Swanson, K. 1994. Just-in-case scheduling. *Proceedings of the 12thNational Conference on Artificial Intelligence (AAAI)* 1098–1104.

Fowler, D. W., and Brown, K. N. 2003. Branching constraint satisfaction problems and markov decision problems compared. *Annals of Operations Research* 118:85–100.

Gao, H. 1995. Building robust schedules using temporal protectionan empirical study of constraint-based scheduling under machine failure uncertainty. *Masters thesis, Department of Industrial Engineering, University of Toronto, Toronto, Canada*.

Hnich, B.; Smith, B.; and Walsh, T. 2004. Dual modelling of permutation and injection problems. *Journal of Artificial Intelligence Research* 21:357–391.

Walsh, T. 2002. Stochastic constraint programming. *Proceedings of 15th European Conference on Artificial Intelligence* 111–115.

Weisstein, E. W. 2006. Normal distribution function. *Mathworld, Wolfram Research, Inc.* http:// mathworld. wolfram. com/ Normal Distribution Function. html.