ASSESSING RISK IN A JOB SCHEDULE: INTEGRATING A SCHEDULING HEURISTIC AND A SIMULATION MODEL TO A SPREADSHEET

Kusuma Rojanapibul
Juta Pichitlamken

Department of Industrial Engineering,
Faculty of Engineering, Kasetsart University
Bangkok 10900, THAILAND

ABSTRACT

Deterministic scheduling algorithms are often applied to problems in stochastic settings perhaps because they are already hard to solve even without considering stochastic characteristics. We are interested in assessing the measure of risk in performance measures (e.g., makespan) when these algorithms are used in probabilistic environment. We design an easy-to-use Microsoft Excel® program that integrates a Visual Basic Application (VBA) subroutine which performs scheduling procedures, with an Arena® simulation model that imitates the stochastic production environment. Our program suggests a job schedule, its associated performance measures and the corresponding prediction intervals. At the moment, we only consider the m-machine permutation flowshop problem with the makespan (or completion time) objective.

1 INTRODUCTION

In practice, deterministic algorithms are often used to solve stochastic scheduling problems perhaps because they are already hard to solve even without considering stochastic characteristics. Consider the permutation flowshop environment where there are n jobs that need to be processed on each of the m machines, $1, 2, \ldots, m$, in that order. At any time, each machine can process at most one job. Once the job starts to be processed, it has to be completed without interruption. Suppose that we are interested in finding the job sequence that minimizes the makespan, defined as the total time to complete all n jobs. Garey, Johnson, and Sethi (1976) shows that this problem with $m \geq 3$ machines and unlimited storage between any two machines is strongly NP-hard.

A deterministic algorithm yields only the expected performance measures but no measure of risk because variability is not considered in the calculation. *The goal of our work is to develop an easy-to-use tool that supplements a deterministic heuristic with some measure of risk.* We do not intend to propose a heuristic for the stochastic flowshop but merely suggest a mean to evaluate a job schedule in a stochastic environment. Some works on the stochastic flowshop include Gourgand, Grangeon, and Norre (2003) and Shaked and Shanthikumar (1994).

Our program not only suggests a job schedule but also provides a prediction interval for the performance measure of interest, say θ. A prediction interval is a measure of risk; it is designed to be wide enough to contain the actual performance measure with high probability. For example, suppose a 95% prediction interval of a makespan for schedule s is (a, b). Next time we use schedule s, the makespan that we experience will fall in (a, b) with 95% probability.

Let the realizations of θ be $Y_1, Y_2, \ldots, Y_R$. The natural estimator of θ is the sample mean, $\bar{Y} = \sum_{i=1}^{R} Y_i / R$. The normal-theory prediction interval is

$$\bar{Y} \pm t_{\alpha/2, R-1} S \sqrt{1 + \frac{1}{R}}, \qquad (1)$$

where $S^2 = \sum_{i=1}^{R} (Y_i - \bar{Y})/(R-1)$, the sample variance across R replications, and $t_{\alpha,\nu}$ is such that $\Pr\{T \leq t_{\alpha,\nu}\} = 1 - \alpha$ for a standard Student's t random variable T with ν degree of freedom.

We consider a prediction interval instead of a confidence interval because a confidence interval is a measure of error whereas a prediction interval is a measure of risk. Theoretically, one can drive the width of a confidence interval to zero by making more replications, but one can not simulate away risk which is inherent in the system (Banks et al. 2005). In this work, the realization Y_i in (1) are outputs from stochastic discrete-event simulation (see Law and Kelton (2000) and Banks et al. (2005) for comprehensive coverage).

We call our application $\mathbf{S_2E}$ (Scheduling and Simulation on Excel) through which a user can:

1. Specify parameters of a scheduling environment (currently, we only consider a flowshop): num-

ber of job types, number of machines, mean and standard deviation of processing times of job j on machine i.

2. Get a job schedule and a makespan from a deterministic scheduling algorithm. In this work, just *Profile Fitting (PF) Heuristic* (Pinedo 2002) is available for a flowshop. Examples of other approaches include Genetic algorithms (Iyer and Saxena 2004) and a branch-and-bound algorithm (Ladhari and Haouari 2005).
3. Get prediction intervals of makespan and finished time of each job from simulation. A user has an option of manipulating the number of simulation replications.

Figure 1 shows the interaction between a spreadsheet program, a scheduler and a simulation model. We choose Excel because it is widely understood, it can accommodate user-defined functions (known as *Macro*), and it can communicate with Arena (Rockwell Software, Inc.), a general purpose simulation modeling environment (Kelton, Sadowski, and Sadowski 2002).

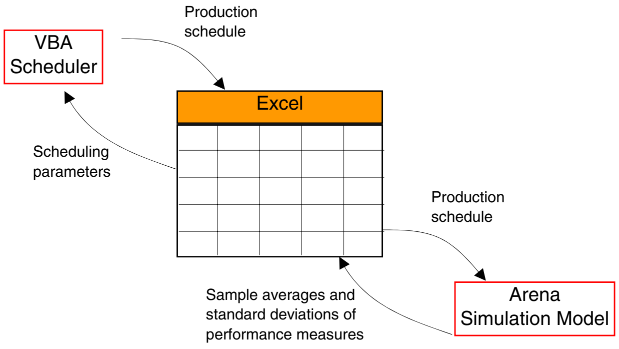


Figure 1: Interaction between Excel, Scheduler and a Simulation Model

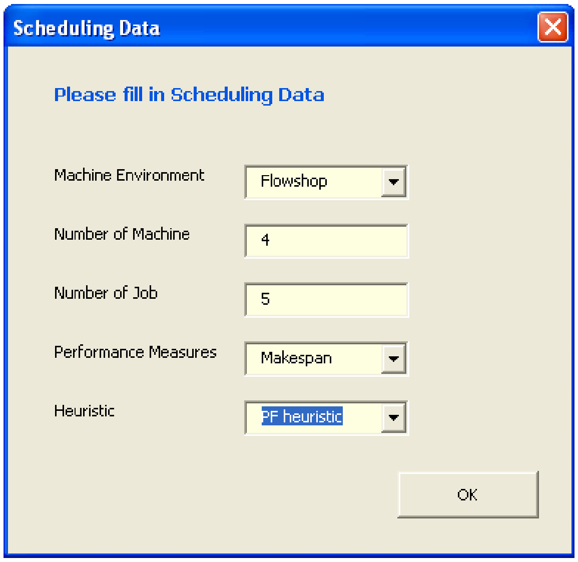
This paper is organized as follows: Section 2 details features of our software. Section 2.1 shows dialogue boxes, inputs and outputs. Section 2.2 describes the scheduling heuristic, and Section 2.3 discusses the simulation model. We provide a numerical example in Section 3. We conclude with future works in Section 4.

2 EXCEL APPLICATION

We explain the S$_2$E interface and what goes beneath in this section.

2.1 Software Inputs and Outputs

After opening up the Excel file, a user is prompted to provide details about a scheduling problem (Figure 2). Figure 3 shows the processing time worksheet where a user needs to specify the mean and standard deviation of processing times of job j on machine i, p_{ij} and σ_{ij}, respectively, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. Then she gets a job schedule by clicking a button “Get a job sequence"; a VBA of the PF heuristic generates a job schedule which is shown on another Excel worksheet (Figure 4), along with the makespan, the departure times of job j from machine i, the wasted times, and the Gantt chart (Figure 5).



Scheduling Data

Please fill in Scheduling Data

Machine Environment: Flowshop

Number of Machine: 4

Number of Job: 5

Performance Measures: Makespan

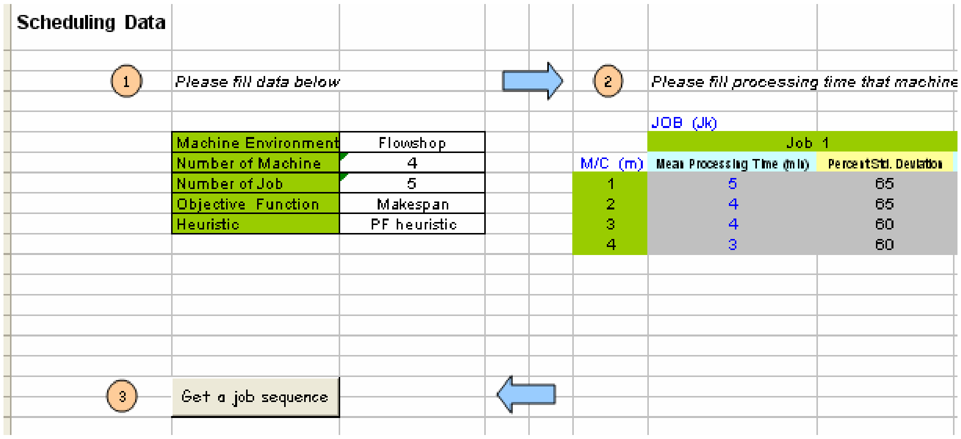
Heuristic: PF heuristic

Figure 2: Input Dialogue Box

A user can get prediction intervals of the performance measures associated with the suggested schedule. By clicking the button “Run simulation," the Arena simulation model of a flowshop is automatically created and executed. When the simulation finishes, the summary results over all simulation replications are shown on a new worksheet (Figure 6). Our software yields prediction intervals of a makespan and finished time of each job. Detailed simulation results are also on display on another worksheet. A user can also launch another set of simulation runs by specifying a confidence level or the width of the prediction interval (Figure 7). Note that the user-specified width of the prediction interval will be rejected if it is too small.

2.2 Profile Fitting Heuristic

The PF heuristic is one of the most popular methods used to minimize the makespan in flowshops with limited storage between stations; thus, blocking—when a job cannot be released to the next station downstream because the downstream buffer is full—can occur. The algorithm proceeds as follows (Pinedo 2002): one job is selected to go first, possibly according to some simple rules, e.g., the job with the smallest sum of processing times (S$_2$E uses this rule).



Scheduling Data

1 *Please fill data below*

Machine Environment	Flowshop
Number of Machine	4
Number of Job	5
Objective Function	Makespan
Heuristic	PF heuristic

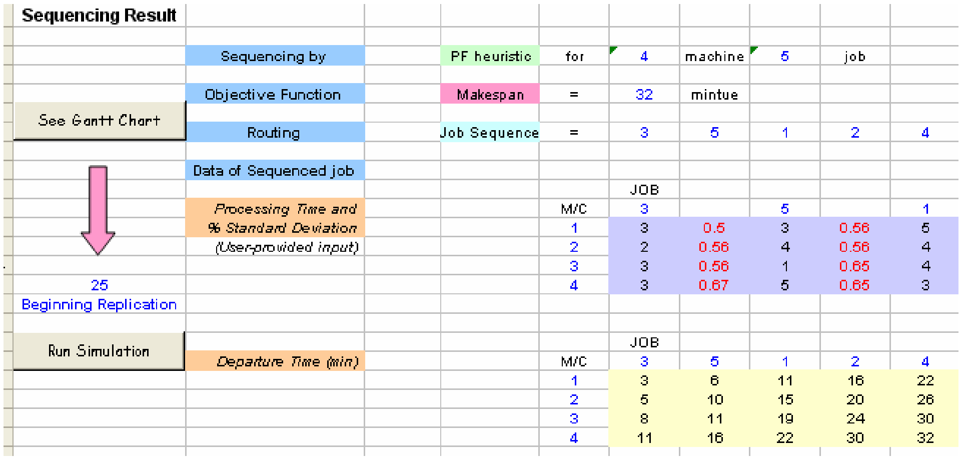
2 *Please fill processing time that machine*

JOB (Jk)

	Job 1	
M/C (m)	Mean Processing Time (min)	Percent Std. Deviation
1	5	65
2	4	65
3	4	60
4	3	60

3 Get a job sequence

Figure 3: Processing Time Worksheet



Sequencing Result

Sequencing by	PF heuristic	for	4	machine	5	job		
Objective Function	Makespan	=	32	mintue				
Routing	Job Sequence	=	3	5	1	2	4	

See Gantt Chart

Data of Sequenced job

Processing Time and % Standard Deviation (User-provided input)

	JOB				
M/C	3		5		1
1	3	0.5	3	0.56	5
2	2	0.56	4	0.56	4
3	3	0.56	1	0.65	4
4	3	0.67	5	0.65	3

25
Beginning Replication

Run Simulation

Departure Time (min)

	JOB				
M/C	3	5	1	2	4
1	3	6	11	16	22
2	5	10	15	20	26
3	8	11	19	24	30
4	11	16	22	30	32

Figure 4: Sequencing Outputs

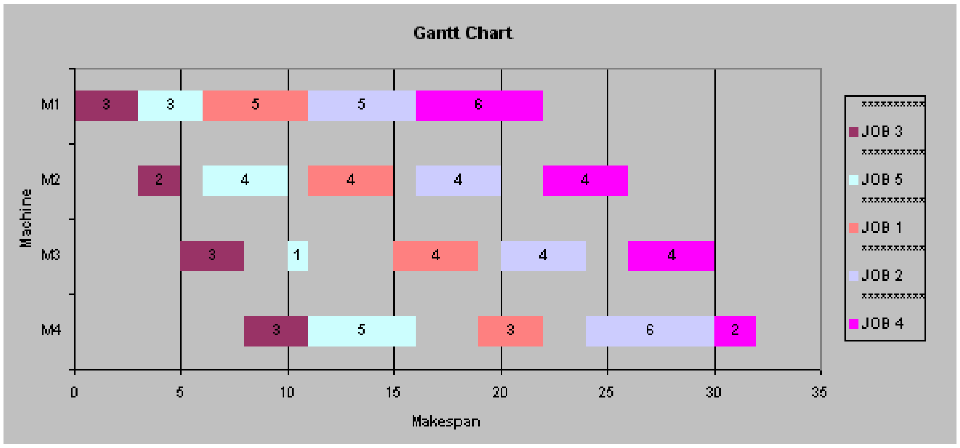
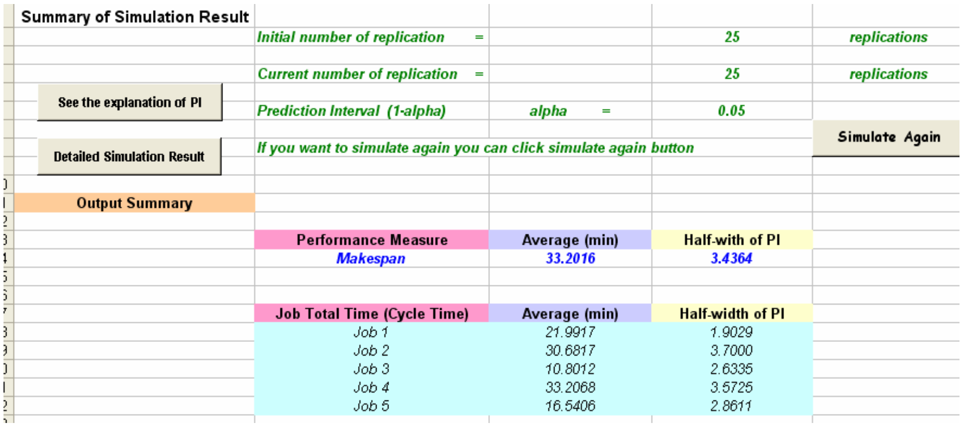


Figure 5: Gantt Chart



Summary of Simulation Result

Initial number of replication = 25 replications

Current number of replication = 25 replications

Prediction Interval (1-alpha) alpha = 0.05

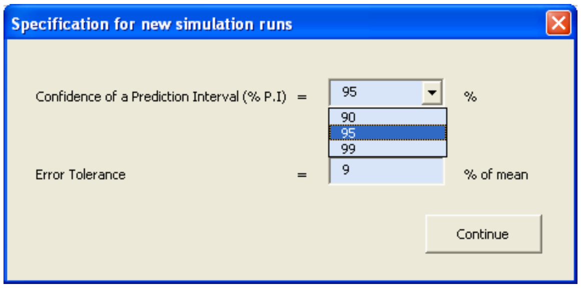
If you want to simulate again you can click simulate again button

Output Summary

Performance Measure	Average (min)	Half-with of PI
Makespan	33.2016	3.4364

Job Total Time (Cycle Time)	Average (min)	Half-width of PI
Job 1	21.9917	1.9029
Job 2	30.6817	3.7000
Job 3	10.8012	2.6335
Job 4	33.2068	3.5725
Job 5	16.5406	2.8611

Figure 6: Summary Simulation Outputs



Specification for new simulation runs

Confidence of a Prediction Interval (% P.I) = 95 %

Error Tolerance = 9 % of mean

Figure 7: Simulation Options for Reruns

This job, say job j_1, does not encounter any blocking. The *profile* of job j_1 is a collection of its departure times from each machine i, i.e., $\{D_{1j_1}, D_{2j_1}, \ldots, D_{mj_1}\}$. If job j_1 corresponds to job k, then

$$D_{ij_1} = \sum_{h=1}^{i} p_{hj_1} = \sum_{h=1}^{i} p_{hk}.$$

To determine which job should go second, every remaining unscheduled job is tried out. For the case when there is no buffer between stations, the departure times of job that goes second, say j_2, can be computed as:

$$\begin{aligned} D_{1j_2} &= \max\left(D_{1j_1} + p_{1j_2}, D_{2j_1}\right) \\ D_{ij_2} &= \max(D_{i-1j_2} + p_{ij_2}, D_{(i+1)j_1}), \\ & \quad i = 2, \ldots, m-1 \\ D_{mj_2} &= \max(D_{(m-1)j_2}, D_{mj_1}) + p_{mj_2}. \end{aligned}$$

Note that a single buffer space can be thought as a machine with zero processing time.

For each job that remains unscheduled, we compute the *wasted* time on machine i—the amount of time that machines are idle and the amount of time the job is blocked—which is simply $D_{ij_2} - D_{ij_1} - p_{ij_2}$. The unscheduled jobs with the smallest wasted time is selected for the second position. This procedure is repeated until all jobs are scheduled. In the method just described, wasted time on all machines have equal weight. A modified PF heuristic where higher weights are placed on the more congested machines has also been proposed (Pinedo 2002).

2.3 Simulation Model

The simulation model uses the PF schedule that we get in Section 2.2 in a flowshop where processing times are probabilistic. We assume that the processing times of job j on machine i are normally distributed with mean p_{ij} and variance $\sigma_{ij}^2, \forall i, j$. The simulation model starts with all machines idle and all jobs ready to be processed. There is no buffer space between any two machines (as noted above, any flowshop with a positive and finite buffer space can be modeled as a flowshop with zero buffer size). A single replication finishes when all n jobs are completed.

From S$_2$E program, our flowshop simulation model is automatically created in Arena 7.01. All communication between Arena and Excel are done through VBA. For example, when simulation finishes, the VBA in Arena reads the output of individual replications from a text file and places it in the original S$_2$E worksheet. The summary statistics are computed inside Excel.

3 NUMERICAL EXAMPLE

We use a simple example of a flowshop with 5 jobs and 4 machines to illustrate potential benefits of simulation experiments. Table 1 presents the mean processing times for this example.

If we select the job that goes in the first position as one with the smallest sum of processing times, and gives equal weight on all machines, the PF heuristic yields the following job sequence: 3, 5, 1, 2 and 4 with the makespan

Table 1: Mean Processing Times, p_{ij}

Machine i	Job j				
	1	2	3	4	5
1	5	5	3	6	3
2	4	4	2	4	4
3	4	4	3	4	1
4	3	6	3	2	5

of 32 time units. We consider three scenarios of different coefficients of variation (defined as standard deviation of a random variable divided by its mean) of processing times: 5%, 10% and 25%. Table 2 shows the sample average makespan from simulation and its prediction interval, PI, (Equation 1) when the number of replication is 25. As expected, when the variability increases, the half width of makespan PI increases. Moreover, the average makespan from simulation, which can be viewed as what we may experience in reality, are also further away from the PF makespan of 32 time units. Of course, intuition tells us that the case with higher variability involves greater risk, but PIs give us tangible risk measures.

Table 2: Prediction Intervals of Makespan at Different Levels of Processing Time Variability

Coefficient of variation	Sample average makespan	Prediction interval		
		Lower bound	Upper bound	Half width
5%	32.08	31.79	32.38	0.59
10%	32.17	31.58	32.76	1.18
25%	32.44	30.96	33.92	2.96

4 FUTURE WORK

We present a decision-aided tool, S$_2$E, which is built on a spreadsheet program that utilizes a deterministic heuristic to generate a job schedule and also makes use of stochastic discrete-event simulation to determine a prediction interval as a measure of risk. We plan to add other algorithms for generating a job schedule so that a user has choices from which she can compare both the mean performance measures and their associated measures of risk.

REFERENCES

Banks, J., J. S. Carson, B. L. Nelson, and D. Nicol. 2005. *Discrete-event system simulation*. 4h ed. Upper Saddle River, NJ: Prentice Hall, Inc.

Garey, M. R., D. S. Johnson, and R. Sethi. 1976. The complexity of flow shop and job shop scheduling. *Mathematics of Operations Research* 1:117–129.

Gourgand, M., N. Grangeon, and S. Norre. 2003. A contribution to the stochastic flow shop scheduling problem. *European Journal of Operational Research* 151:415–433.

Iyer, S. K., and B. Saxena. 2004. Improved genetic algorithm for the permutation flowshop scheduling problem. *Computers & Operations Research* 31:593–606.

Kelton, D. W., R. P. Sadowski, and D. A. Sadowski. 2002. *Simulation with Arena*. 2d ed. New York: McGraw-Hill.

Ladhari, T., and M. Haouari. 2005. A computational study of the permutation flow shop problem based on a tight lower bound. *Computers & Operations Research* 32 (7). 1831–1847.

Law, A. M., and W. D. Kelton. 2000. *Simulation modeling and analysis*. 3d ed. Singapore: McGraw-Hill.

Pinedo, M. L. 2002. *Scheduling theory: Algorithms and system*. 2d ed. New Jersey: Prentice-Hall.

Shaked, M., and J. G. Shanthikumar. 1994. *Stochastic orders and their applications*. Boston: Academic Press.

AUTHOR BIOGRAPHIES

KUSUMA ROJANAPIBUL is a graduate student in the Department of Industrial Engineering, Kasetsart University. Her summer internship at Siam Cement (Plc) Ltd., in 2004 got her interested in scheduling and simulation. Her e-mail address is <gift_xx@yahoo.com>.

JUTA PICHITLAMKEN is a Lecturer in the Department of Industrial Engineering, Kasetsart University. Her research interests include ranking and selection procedures and simulation optimization. Her e-mail address is <juta.p@ku.ac.th>, and her web page is <http://pirun.ku.ac.th/~fengjtp/>.