International Journal of Engineering Technologies and Management Research

IJETMR

A Knowledge Repository

JOB-SHOP SCHEDULING OPTIMIZATION WITH STOCHASTIC PROCESSING TIMES

Jaber S. Alzahrani [*1]

[*1] Department of Industrial Engineering, Engineering College at Alqunfudah, Umm Al-Qura University, Saudi Arabia

Abstract:

In this study, a job shop scheduling optimization model under risk has been developed to minimize the make span. This model has been built using Microsoft Excel spreadsheets and solved using @Risk solver. A set of experiments have been also conducted to examine the accuracy of the model and its effectiveness has been proven.

Keywords: JSSP; Genetic Algorithm; Optimization; @Risk.

1. Introduction

Job Shop Scheduling (JSS) problems with random processing times under various costs have been considered by researchers for its subsequent costs of the uncompleted job on its due time, to minimize the average value of its total costs. The general procedure of heuristic decision-making rules is used in situations when several jobs are ready to be served on one machine [4-6]. Flow Shop Scheduling (FSS) with random processing times has been studied by Singer [3] to minimize the expected total weighted tardiness. Moreover, a simulation-based genetic algorithm for solving a job shop with random processing times has been proposed by Yoshitomi [7] in order to minimize the expected makespan. Yoshitomi and Yamaguchi [8] have enhanced this genetic algorithm by applying a new cross over operator. In these two algorithms, solutions having very high frequency through all generations are selected as good solutions. Tavakkoli-Moghaddam et al. [9] have also considered a job shop with random operations, where the time difference between the delivery and completion of jobs as well as related operational or idle cost of machines that must be minimized. They have presented a hybrid method consisting of a neural network to generate an initial solution and a simulated annealing algorithm to improve the quality of the initial solution. Some authors have developed exact and heuristic algorithms for job shop scheduling problems with the makespan or mean flow time criterion subject to random processing times, which can take any real value between known lower and upper bounds [10-12]. Luh et al. [13] have presented a job shop approach based on a combined Lagrangian relaxation and stochastic dynamic programming by taking into consideration uncertain arrival times, processing times, due dates, and part priorities, in order to achieve minimizing of expected job tardiness and earliness cost. Neumann and Schneider [14] have proposed heuristics based on aggregate schedules for a job shop scheduling problem with stochastic precedence constraints, where the expected makespan is to be minimized.

Finally, Alcaide et al. [15] have studied open shop scheduling problems of random processing times, breakdowns and repair times, where the expected makespan must be minimized. They developed a general dynamic procedure that converts any problem into a set of scheduling problems without breakdowns.

In the recent literature, two new criteria have been brought to the attention of researchers for their consideration: robustness and stability [1]. Robustness means that schedule performance is still acceptable when something unforeseen happens. However, Stability means that schedule does not deviate significantly due to disruption and revision. In general, there are two methodologies to deal with the uncertainty in a scheduling environment: proactive and reactive scheduling. Incorporating the knowledge of uncertainty at the decision stage, proactive approaches focus on generating more robust predictive schedules to minimize the effects of disruptions. On the other hand, reactive scheduling algorithms are implemented at execution time to adjust the schedule according to the real-time situation when the uncertainty is realized, or disruptions occur. This paper proposes a proactive approach to generate a long-term initial schedule under uncertainty to jointly determine the production planning and preventive maintenance to simultaneously optimize system bi-objective of robustness and stability.

Zhiqiang Lu et al. [2] addressed the problem of finding a robust and stable schedule for a single machine with availability constraints. The machine suffers unexpected breakdowns and follows the Weibull failure function.

The mission of the scheduling process is to optimally allocate the suitable equipment to perform the required jobs over a period to achieve the business goals. Therefore, many efforts have been devoted to solve most optimal Job Shop Scheduling Problems (JSSP), as most of the researches aimed at minimizing the maximum completion time. JSSP is an NP-hard problem; so, it is difficult to find an exact solution in a reasonable computation time [16]. Number of optimization methods have been developed to solve JSSP, Tabu Search [17, 18], Simulated Annealing [19, 20], Genetic Algorithms [21, 22], Particle Swarm Optimization [23, 24], Ant Colony Optimization [25, 26], differential evolution algorithm [27], Memetic Algorithm [28], Mathematical Programming [29, 30], and Goal Programming [31].

Genetic Algorithm has the advantage to solve scheduling problems. It can reach optimal or sub-optimal solutions. It has gained the advantage of getting the global optimization solution in a huge population.

Risk analysis in @RISK is known as a quantitative method that seeks to determine the probability distributions of outcomes resulting from decisions. In general, the techniques in a @RISK risk analysis encompass four steps:
1) **Developing a Model**. Define your problem or situation in an Excel model.
2) **Identifying Inputs and Outputs**. Determine uncertain inputs in the model, specify their possible values with probability distributions and identify the important outputs you want to analyze.
3) **Analyzing the Model with Simulation**. Run many scenarios with sampled values in each one for the uncertain inputs, to determine the probability distributions of your outputs.
4) **Deciding**. Use the simulated results and personal preferences to make informed choices.

The first three steps of @RISK help in providing a powerful and flexible set of tools in Excel to facilitate model building and risk analysis. The results generated by @RISK can then be used by a decision maker to choose a course of action.

The aim of this paper is to present a job shop scheduling (JSS) optimization model under risk is developed to minimize the makespan. The model has been built using the Microsoft Excel spreadsheets and solved using @Risk.

2. Problem Description and Assumptions

The goal of this model is to Solve a JSS problem where spreadsheet-based commercial genetic algorithm solver "@Risk" [32] is used to optimize the makespan function in below mentioned Equation 4.

Classical Job Shop Scheduling Problems (JSSP) considers the allocation of n jobs to m different machines or equipment. Each job has to undergo multiple operations in various equipment, with its own set of processing times and routing characteristics. The processing time of each job on an equipment P_{hj} is known as well as the due date for each job D_j.

The model assumes the following:
- The processing times are stochastic;
- Each job has its own due date;
- Each job will not visit "undergone" by the same equipment twice;
- All jobs are ready for processing at time zero;
- All equipment are available at time zero;
- Each equipment can process only one job at a time;
- Only one job can be processed by an equipment at any instant in time;
- Set-up time for any operation is included in the processing time;
- The required transportation time for the movement of jobs between equipment is assumed to be negligible;
- The operation couldn't be interrupted;
- There are no precedence constraints among operations of different jobs.

3. Model Formulation

Parameters
N: Number of jobs
M: Number of machines
P_{ji}: Processing time for job j on m/c, j = 1, 2, …, N and i = 1, 2, …, M
D_j: Due date of job j, j =1, 2, …, N
SEQ: Processing sequence array
NUMT: No. of machines (tasks) for each job
NUMJ: No. of jobs per machine J
DISJ: Disjunction array.

Decision Variables
C_j: Completion time of job j
S_{ji}: Starting time of job j on machine i
F_{ji}: Finishing time of job j on machine i
P_{ji}: Processing time for job j on m/c, j = 1, 2, …, N and i = 1, 2, …, M

3.1. Objective Functions

The objective of this model is to minimize the makespan which has been defined in Equation 1.

$$\text{Makespan} = \text{MAX}\left(C_j\right), \forall j \in N \tag{1}$$

3.2. Constraints

1) Disjunction constraints (to avoid overlapping between tasks)

$$\left(S_{hi} - S_{hj}\right) \geq P_{hj} - M\, Y_{hij}\,, \forall i, j \in N, \forall h \in M \tag{2}$$

$$\left(S_{hj} - S_{hi}\right) \geq P_{hi} - M\,(1 - Y_{hij})\,, \forall i, j \in N, \forall h \in M \tag{3}$$

2) Conjunction constraints (to satisfy operational precedence)

$$\sum_{h \in M}\left(S_{SEQ(j,l),j} + P_{SEQ(j,l),j}\right) \geq \sum_{h \in M} S_{SEQ(j,l+1),j}\,, \forall j \in N, \forall l \in M - 1 \tag{4}$$

4. Computational Results and Analysis

In this section, the results of the proposed model application are introduced. In consequence, the model has been solved using @Risk optimizer which operates in an Intel® Core™ i3-2310M CPU @2.10 GHz (3 GB of RAM). The GA parameters include; population size N = 50, number of generations G = 40,000, probability of crossover Pc = 0.5, and probability of mutation Pm = 0.1.

To avoid doubt, it is meaningful to distinguish between the terms "trials" and "iterations" in the RISK Optimizer process. The overall process involves a sequence of "trials," where a given set of adjustable cell values is tested on each trial. For the evaluation of the target cell (and the constraint cells) for a trial, a standard @RISK simulation is run for a specified number of "iterations." This distinction helps in understanding why RISK Optimizer can take some time. For example, it might require 1000 trials to converge to optimality, and each of these trials might require a 500-iteration simulation. There will be a lot of computer calculation, especially for complex models.

As in traditional optimizers, constraints can be entered in RISKOptimizer. When the value of the constrained cell does not change during a simulation, the constraint is called non-probabilistic. Otherwise, it is called probabilistic. A probabilistic constraint is based on a statistic of the distribution of the cell's values, such as "the mean of A1 <= 100." RISKOptimizer calculates the statistic at the end of each simulation to decide whether the constraint is satisfied or not.
The model accuracy and capability are verified by solving and analyzing a 5J*4M problem.
The model inputs, processing sequences, and durations are assumed as shown in Tables 1.

Table 1: Job's processing sequences

Job	Processing Sequence (Duration)			
J1	M1 (19)	M3 (10)	M4 (19)	
J2	M1 (10)	M2 (30)	M3 (18)	M4 (10)
J3	M4 (15)	M3 (18)	M2 (31)	
J4	M2 (14)	M3 (10)	M4 (20)	M1 (19)
J5	M1 (15)	M3 (16)		

Initially, the problem has been solved as a deterministic to set the base of comparison and discussion for the stochastic solutions. The Gantt chart of the deterministic case is shown in Figure 1.
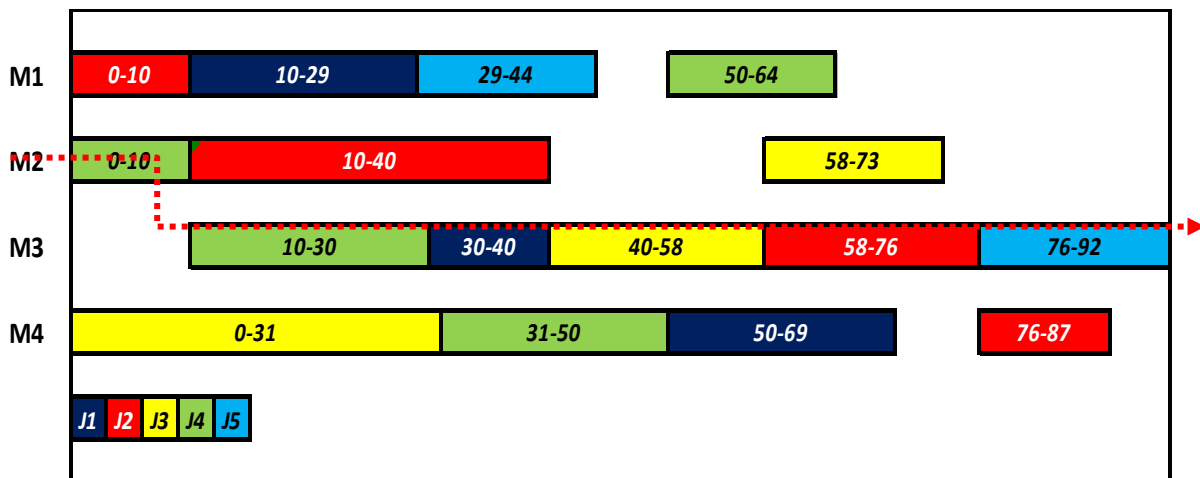
Figure 1: Gantt chart of the makespan objective

In Figure 1, the dotted red line represents the critical path operations which include operations O42, O43, O13, O23, O33, and O53.

It may be noticed in the Gantt chart that there is a slack time of 18 units of time after finishing the second operation of the second job on the second machine. So that, the effect of changing the processing time of this operation into stochastic for the optimal makespan will be studied.

The duration P_{22} is assumed to follow a uniform distribution among a minimum value 27, a maximum value 33 and random value as shown in Figure 2.
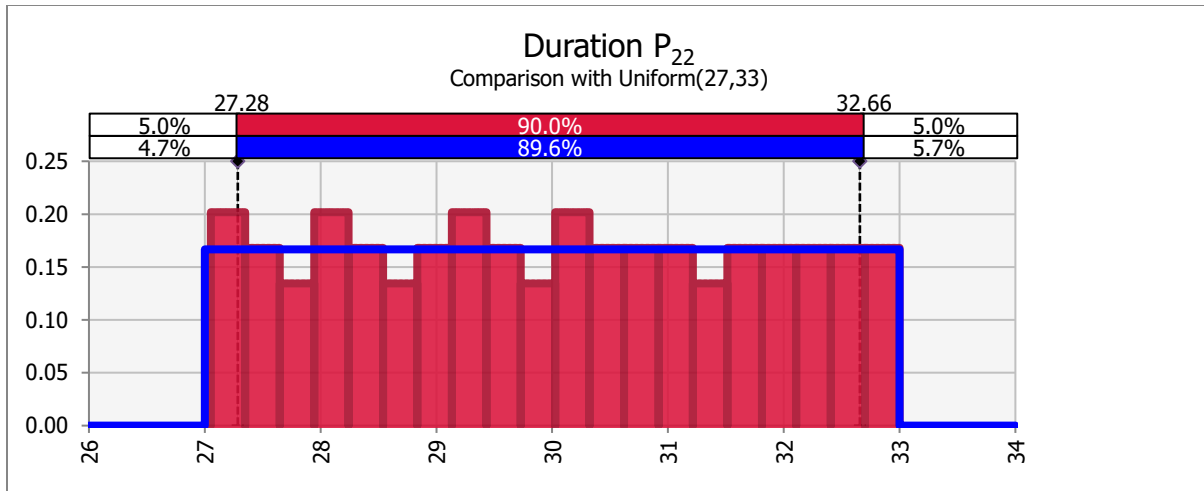
Figure 2: Duration distribution of job 2 on machine 2

Figure 3, shows operation finishing time in which it has a uniform distribution. As this operation is not a critical one form those shown in Figure 1, the optimal makespan remains the same of about 92. It was in the deterministic case as shown in Figure 4.
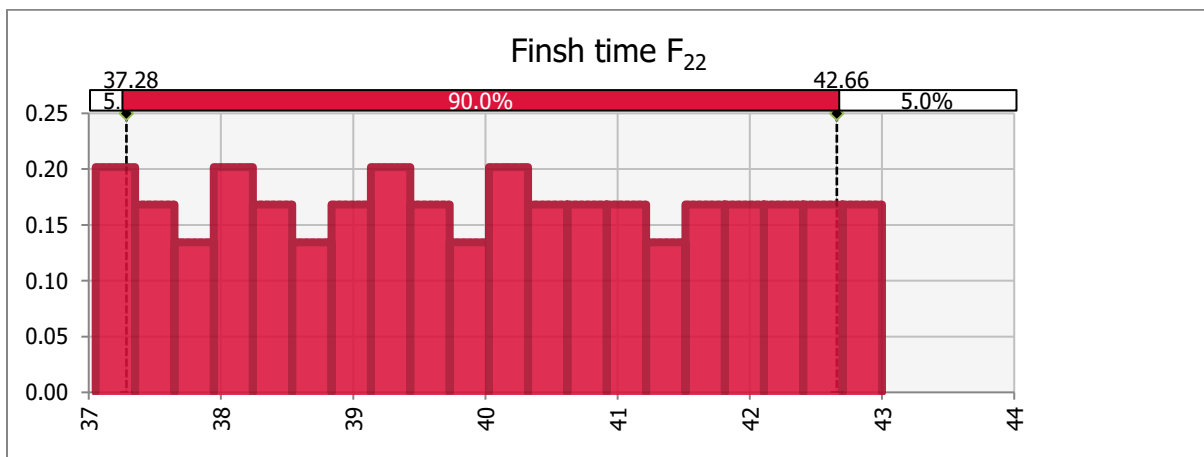
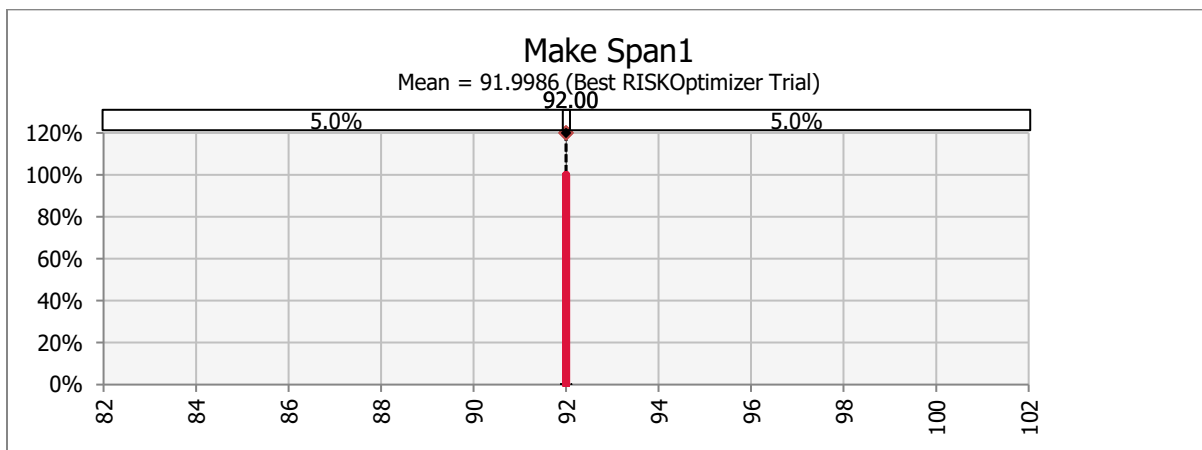Figure 3: Finishing time distribution of job 2 on machine 2

Figure 4: Makespan optimal distribution due to the uncertainty of job 2 duration on machine 2

It may be noticed in the Gantt chart in Figure 1 that the second operation of job 5, machine 3 is one of the critical operations. Therefore, this experiment has to study the effect of changing the processing time of this operation into stochastic for the optimal makespan. The duration P_{53} of this critical operation is assumed to follow a uniform distribution of a minimum value by 14, a maximum value by 18 and random value as shown in Figure 5.
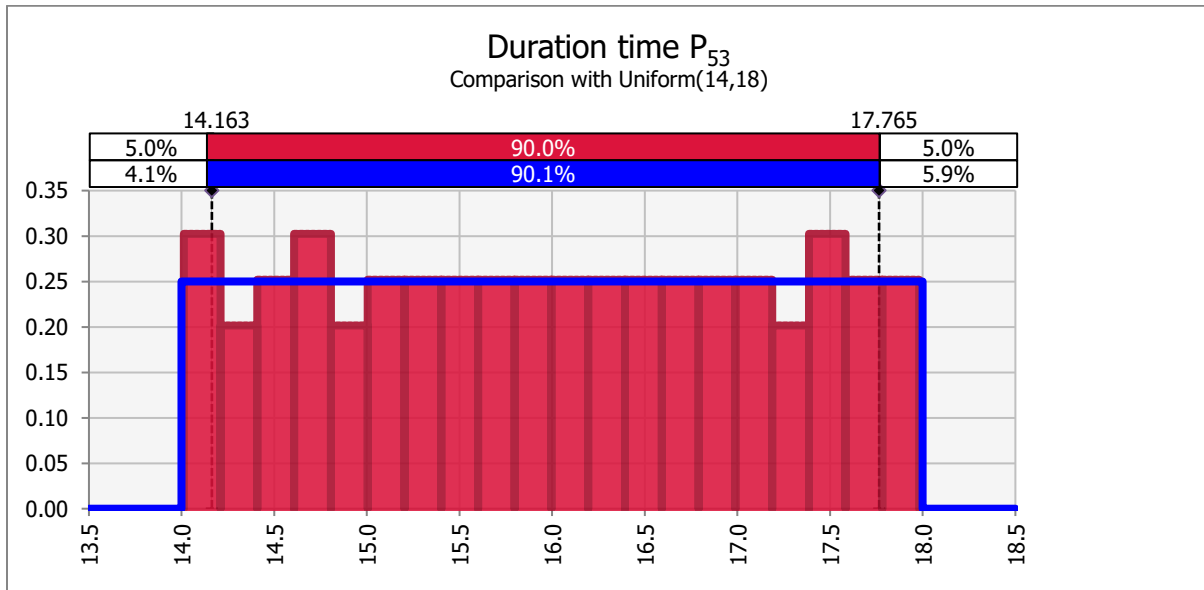

Figure 5: Duration distribution of job 5 on machine 3

Figure 6 shows the operation finishing time in which it has a uniform distribution shape. As this operation is one of the critical operations, the optimal makespan will not have a single value as in the previous case but it will have a probabilistic distribution of uniform shape as shown in Figure 7.
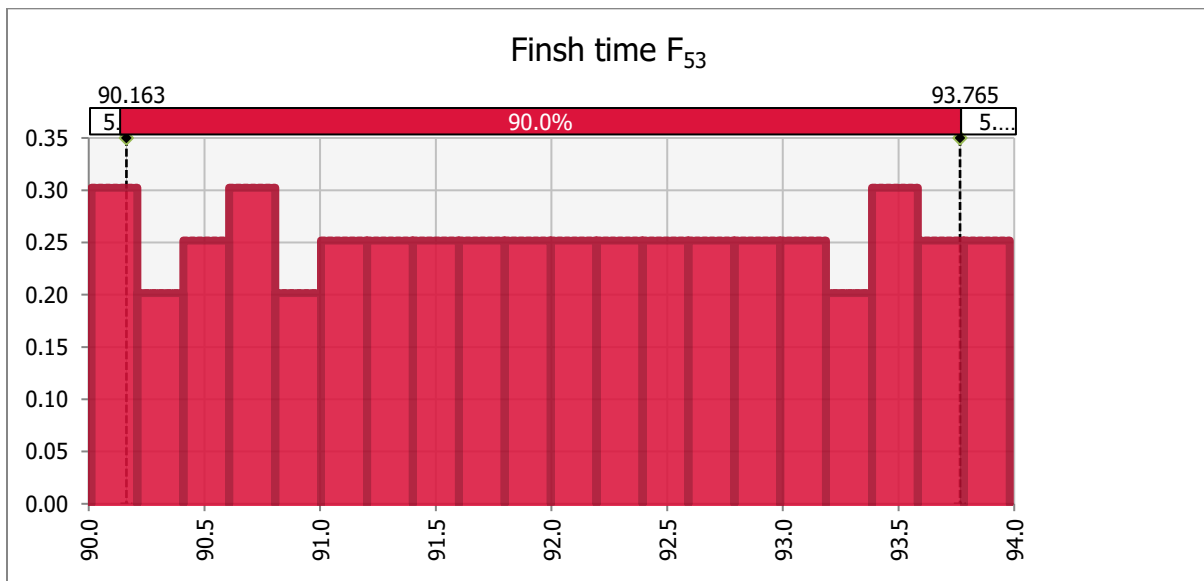

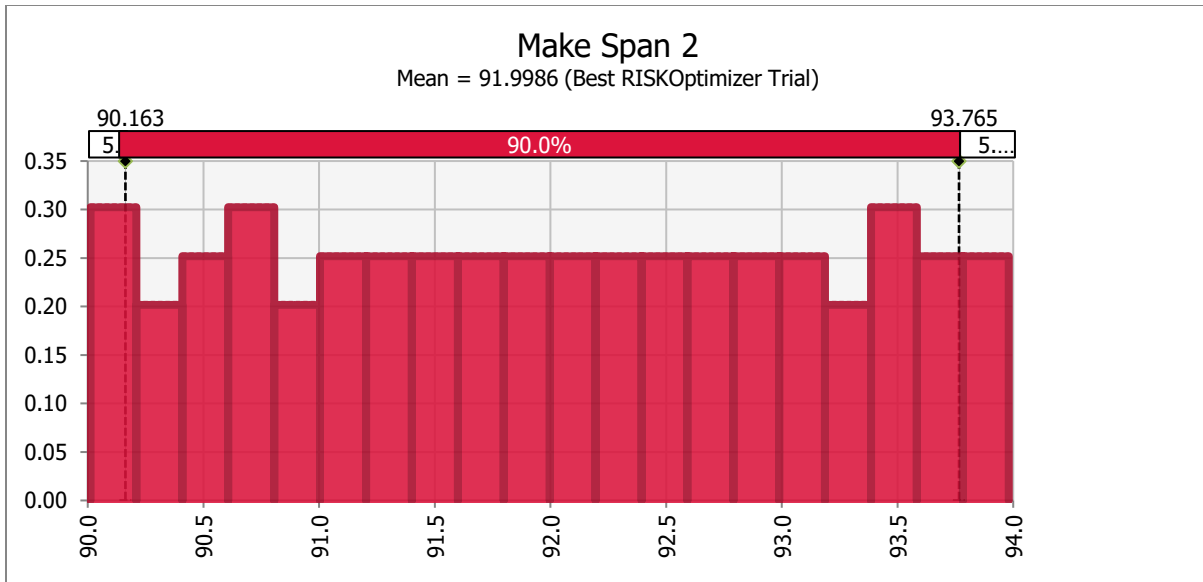Figure 6: Finishing time distribution of job 5 on machine 3

Figure 7: Makespan optimal distribution due to uncertainty of job 5 duration on machine 3

In this experiment, the processing times of all jobs are assumed to be stochastic and follow a normal distribution. Since the normal distribution has two parameters, the standard deviation is given as a ratio to the mean; the ratio of the standard deviation to the mean is known as the variability and assumed to be 10%; N (μ_{ij}, σ_{ij}) where σ_{ij} equals 10 % of μ_{ij}.

Figure 8 shows the distribution of resulted optimal makespan by 10% variability. In addition, Figure 9 depicts the correlation between the resulted makespan and all other uncertain processing times in which it is obvious that the most correlated operations to the makespan are those of O53 and O42 which represent the first and last operations of the critical path.
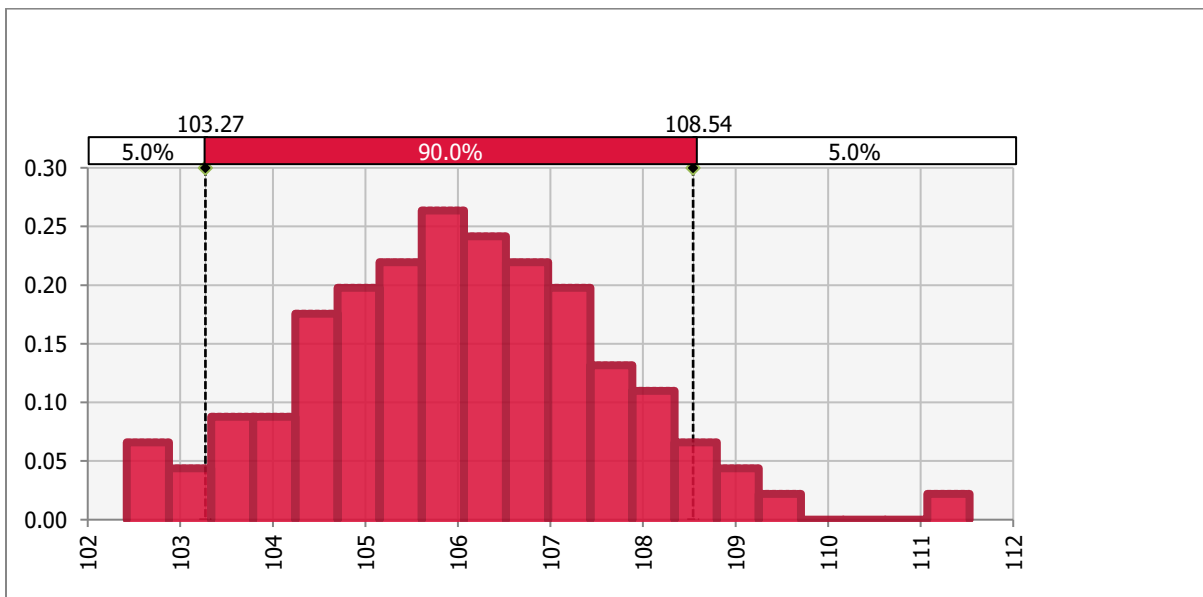

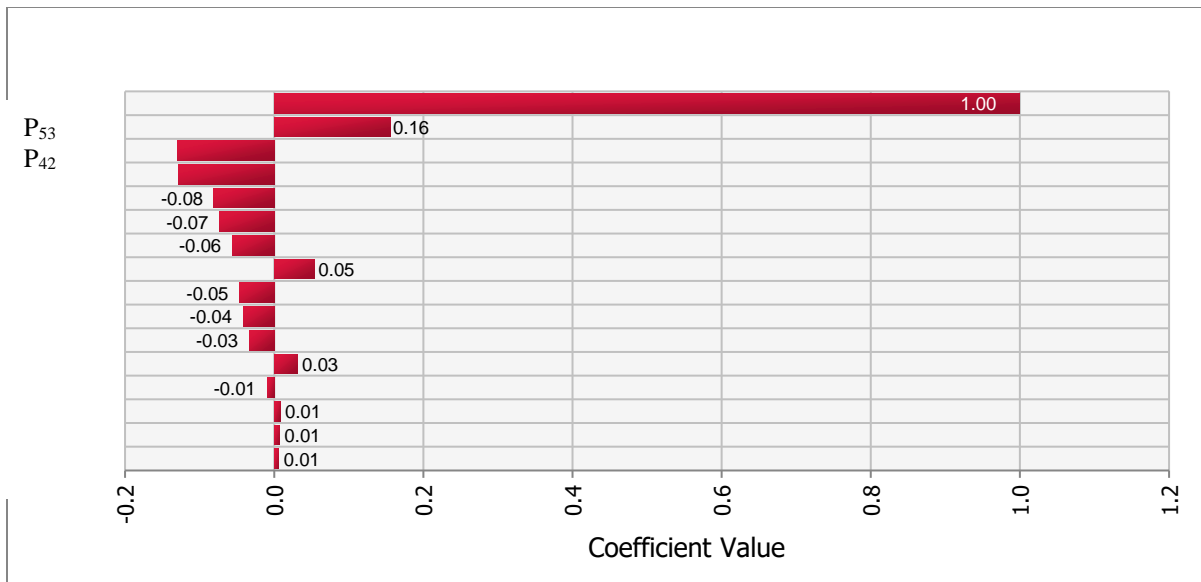Figure 8: Makespan optimal distribution due to 10% variability

Figure 9: Correlation Coefficients (Spearman Rank)

In Figure 10, the change in makespan mean across a range of uncertain processing times is studied and presented. It is noticed that duration P53 has the greatest effect on the makespan which may assure the last conclusion in Figure 9.
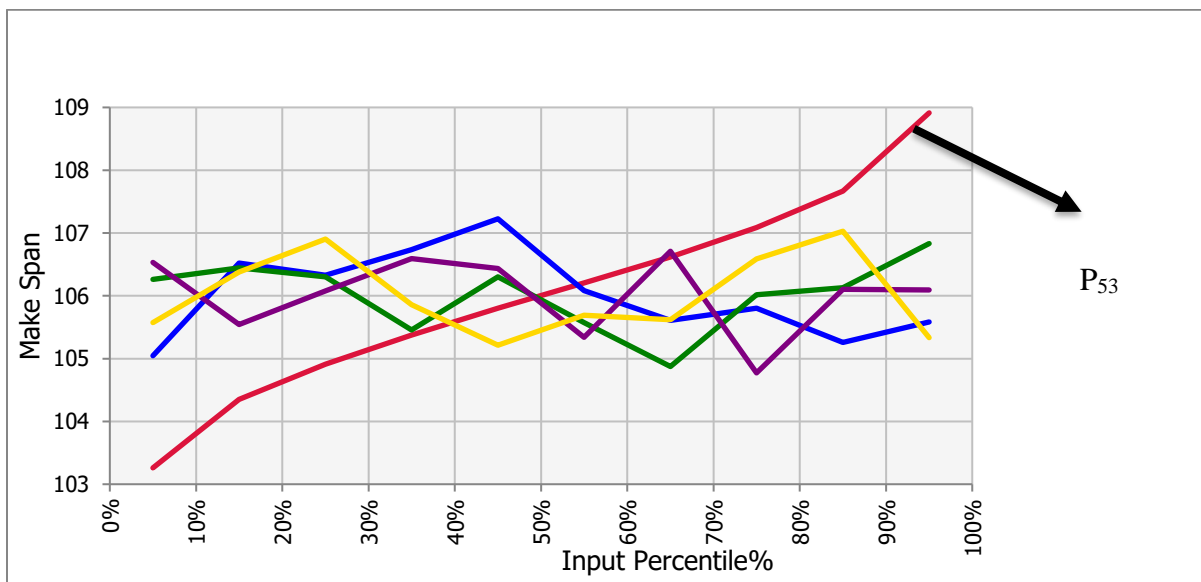
Figure 10: Change in makespan mean Across Range of Input Values

5. Conclusion

The JSS optimization problem has been successfully solved to optimize the makespan under the situation of the uncertainty of the processing times of all jobs.

A developed JSS optimization model has been built using the Microsoft Excel spreadsheets and solved using @Risk solver. A set of four experiments have been conducted to solve the problem

using deterministic processing times, uncertain non-critical operation processing time, uncertain critical operation processing time and uncertainty of all operations processing times.

Finally; the accuracy of the model and its effectiveness have been proven through the analysis of the obtained results.

References

[1] S. Goren and I. Sabuncuoglu, "Robustness and stability measures for scheduling: single-machine environment," IIE Transactions, vol. 40, pp. 66-83, 2008.

[2] Z. Lu, W. Cui, and X. Han, "Integrated production and preventive maintenance scheduling for a single machine with failure uncertainty," Computers & Industrial Engineering, vol. 80, pp. 236-244, 2015.

[3] M. Singer, "Forecasting policies for scheduling a stochastic due date job shop," International Journal of Production Research, vol. 38, pp. 3623-3637, 2000.

[4] D. Golenko-Ginzburg, S. Kesler, and Z. Landsman, "Industrial job-shop scheduling with random operations and different priorities," International journal of production economics, vol. 40, pp. 185-195, 1995.

[5] D. Golenko-Ginzburg and A. Gonik, "Using "look ahead" techniques in job-shop scheduling with random operations," International journal of production economics, vol. 50, pp. 13-22, 1997.

[6] D. Golenko-Ginzburg and A. Gonik, "Optimal job-shop scheduling with random operations and cost objectives," International Journal of Production Economics, vol. 76, pp. 147-157, 2002.

[7] Y. Yoshitomi, "A genetic algorithm approach to solving stochastic job-shop scheduling problems," International Transactions in Operational Research, vol. 9, pp. 479-495, 2002.

[8] Y. Yoshitomi and R. Yamaguchi, "A genetic algorithm and the Monte Carlo method for stochastic job-shop scheduling," International Transactions in Operational Research, vol. 10, pp. 577-596, 2003.

[9] R. Tavakkoli-Moghaddam, F. Jolai, F. Vaziri, P. Ahmed, and A. Azaron, "A hybrid method for solving stochastic job shop scheduling problems," Applied mathematics and computation, vol. 170, pp. 185-206, 2005.

[10] T.-C. Lai, Y. N. Sotskov, N. Y. Sotskova, and F. Werner, "Optimal makespan scheduling with given bounds of processing times," Mathematical and Computer Modelling, vol. 26, pp. 67-86, 1997.

[11] T. Lai and Y. N. Sotskov, "Sequencing with uncertain numerical data for makespan minimisation," Journal of the Operational Research Society, vol. 50, pp. 230-243, 1999.

[12] T.-C. Lai, Y. N. Sotskov, N. Sotskova, and F. Werner, "Mean flow time minimization with given bounds of processing times," European Journal of Operational Research, vol. 159, pp. 558-573, 2004.

[13] P. B. Luh, D. Chen, and L. S. Thakur, "An effective approach for job-shop scheduling with uncertain processing requirements," IEEE Transactions on Robotics and Automation, vol. 15, pp. 328-339, 1999.

[14] K. Neumann and W. G. Schneider, "Heuristic algorithms for job-shop scheduling problems with stochastic precedence constraints," Annals of Operations Research, vol. 92, pp. 45-63, 1999.

[15] D. Alcaide, A. Rodriguez-Gonzalez, and J. Sicilia, "A heuristic approach to minimize expected makespan in open shops subject to stochastic processing times and failures," International Journal of Flexible Manufacturing Systems, vol. 17, pp. 201-226, 2005.

[16] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," Mathematics of operations research, vol. 1, pp. 117-129, 1976.

[17] M. Dell'Amico and M. Trubian, "Applying tabu search to the job-shop scheduling problem," Annals of Operations research, vol. 41, pp. 231-252, 1993.

[18] A. Ponsich and C. A. C. Coello, "A hybrid differential evolution—tabu search algorithm for the solution of job-shop scheduling problems," Applied Soft Computing, vol. 13, pp. 462-474, 2013.

[19] H. R. Lourenco, "Job-shop scheduling: Computational study of local search and large-step optimization methods," European Journal of Operational Research, vol. 83, pp. 347-364, 1995.

[20] M. Faccio, J. Ries, and N. Saggiorno, "Simulated annealing approach to solve dual resource constrained job shop scheduling problems: layout impact analysis on solution quality," International Journal of Mathematics in Operational Research, vol. 7, pp. 609-629, 2015.

[21] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms—I. Representation," Computers & industrial engineering, vol. 30, pp. 983-997, 1996.

[22] Y. Wang, "A new hybrid genetic algorithm for job shop scheduling problem," Computers & Operations Research, vol. 39, pp. 2291-2299, 2012.

[23] D. Sha and C.-Y. Hsu, "A hybrid particle swarm optimization for job shop scheduling problem," Computers & Industrial Engineering, vol. 51, pp. 791-808, 2006.

[24] T.-L. Lin, S.-J. Horng, T.-W. Kao, Y.-H. Chen, R.-S. Run, R.-J. Chen, et al., "An efficient job-shop scheduling algorithm based on particle swarm optimization," Expert Systems with Applications, vol. 37, pp. 2629-2636, 2010.

[25] K.-L. Huang and C.-J. Liao, "Ant colony optimization combined with taboo search for the job shop scheduling problem," Computers & operations research, vol. 35, pp. 1030-1046, 2008.

[26] A. Udomsakdigool and V. Kachitvichyanukul, "Multiple colony ant algorithm for job-shop scheduling problem," International Journal of Production Research, vol. 46, pp. 4155-4175, 2008.

[27] P. Quanke, W. Ling, and G. Liang, "Differential evolution algorithm based on blocks on critical path for job shop scheduling problems," Journal of Mechanical Engineering, vol. 46, pp. 182-188, 2010.

[28] L. Gao, G. Zhang, L. Zhang, and X. Li, "An efficient memetic algorithm for solving the job shop scheduling problem," Computers & Industrial Engineering, vol. 60, pp. 699-705, 2011.

[29] M. S. Al-Ashhab, Munshi, S., Oreijah, M., & Ghulman, H., "Job Shop Scheduling Using Mixed Integer Programming," International Journal of Modern Engineering Research, vol. 7, p. 7, 2017.

[30] K. R. Baker and B. Keller, "Solving the single-machine sequencing problem using integer programming," Computers & Industrial Engineering, vol. 59, pp. 730-735, 2010.

[31] M. Al-Ashhab, "Multi-Objective Job Shop Scheduling Using a Lexicographic Procedure."

[32] N. Palisade, "Guide to Evolver–The genetic algorithm solver for Microsoft Excel," ed: Newfield, NY: Palisade Corporation, 1998.

*Corresponding author.
E-mail address: jszahrani@ uqu.edu.sa