International Journal of Business and
General Management (IJBGM)
ISSN (P): 2319-2267; ISSN (E): 2319-2275
Vol. 8, Issue 3, Apr - May2019; 23-38

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

PENALTY MINIMIZATION JOB SHOP SCHEDULING UNDER UNCERTAINTY

M. S. Al-Ashhab[1], Taiser Attia[2] & Amr Shaaban[3]

[1]Research Scholar, Department of Design & Production Engineering, Faculty of Engineering, Ain-Shams University, Cairo, Egypt and Department of Mechanical Engineering, College of Engineering and Islamic Architecture, Umm Al-Qura University, Makkah, Saudi Arabia

[2,3] Research Scholar, Department of Design & Production Engineering, Faculty of Engineering, Ain-Shams University, Cairo, Egypt

ABSTRACT

In this study, a penalty minimization job-shop scheduling model under uncertainty is developed. The model considers a job-shop of J jobs and M machines. Each task has a random duration with a specific probability distribution. Each job has a specific due date and the bulk penalty if it is not delivered on time. An additional penalty must be paid for each time unit of delay.If any job is accomplished early, it will cost holding expenses. The problem is to determine the optimal start times of each task to minimize the expected penalties. A numerical problem has been solved to minimize both the makespan and total penalties separately and a comparison between results was done. Analysis of the results prescribed that optimizing penalties is important to be taken into consideration besides considering the uncertainty in JSSP.

KEYWORDS: *Job-Shop Scheduling; Optimization; Uncertainty; @Riskoptimizer*

Article History

Received: 04 May 2019| Revised: 08 May 2019| Accepted: 21 May 2019

INTRODUCTION

Scheduling plays an essential role in production systems. This paper considers job-shop scheduling problem (JSSP) where it is the most general and practical machine environment in which several machines exist, and each job has its own pre-determined route to follow. A job shop is defined as any system that processes several tasks for the number of jobs on several machines. Reaching the optimal solution is not an easy task. Till now, no known algorithms guarantee to give an optimal solution and run in polynomial time, consequently optimizing methods such as branch and bound, dynamic programming and mathematical models may be used for small size problems. So, efforts are diverted to heuristic and metaheuristic techniques and algorithms opting to reach near-optimal solutions.

Most of the researchers aimed at optimizing time-based objectives rather than the cost-based objectives. Moreover, they tackled the problem with the assumption of constant job operation times while others consider the randomness of time. Metaheuristic approaches such as genetic algorithm, simulating annealing SA, bee colony BC, ant colony AC, particle swarm optimization PSO, cuckoo nest search CS and migrating bird's optimization MBO and others are presented as solution approaches to JSSP. Banu Calis et al. [1] summarized the well-known objective functions of JSSP and most of the AI used solution strategies to solve the problem.

S.Singh et al. [2] proposed a hybrid algorithm using cuckoo search optimization CSO with enhancement scheme to solve the problem with an objective of minimizing makespan.

Bing Wang et al. [3], established a hybrid local search robust optimization model combining the tabu technique and the simulated-annealing search. They considered uncertain processing times with makespan as a performance criterion. They described the uncertain processing times by discrete scenarios. The set of bad scenarios hedges against the uncertainty of achieving substandard performances among these bad scenarios.

The mission of the scheduling process is to optimally allocate the suitable machine to perform the required jobs over a period to achieve the business goals. Therefore, many efforts have been performed to solve most optimal JSSP, as most of the researches aimed at minimizing the maximum completion time. Number of methods have been developed to solve JSSP; Tabu-Search[4], Simulated Annealing[5], Genetic Algorithms[6], Particle Swarm Optimization [7, 8], Ant Colony Optimization [9, 10], differential evolution algorithm [11], Memetic Algorithm [12], Mathematical Programming [13, 14] and Goal Programming [15].Some researchers managed the problem under uncertainty analysis. RuhulSarker et al. [16] considered the disruption problem of receiving different job orders from customers and facing frequent machine breakdowns.

Tavakkoli-Moghaddam et al. [17] considered JSS 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. Some authors have developed exact and heuristic algorithms for JSSP with the makespan or mean flow time criterion subject to random processing times [18-20].

In recent literature, two new criteria have been brought to the attention of researchers for their consideration: robustness and stability [21].Z. Lu et al. [22] addressed the problem of finding a robust and stable schedule for a single machine with availability constraints.

Chan et al. [23, 24]considered in their model, the minimization of late cost, inventory cost, penalty cost, setup cost besides makespan. While Huang [25] considered minimizing the material processing cost, setup time cost and inventory cost as their objective function.Varthanan et al. [26] developed an efficient particle swarm algorithm to solve both deterministic and stochastic problems and minimizing the total cost. D. Golenko et al. [27] developed an optimization model to solve JSSP with random durations and various cost penalties and expenses.

From the aforementioned, most of the developed heuristics in JSSP targeted the problem of time to minimize makespan. Less work was done to consider the role of cost on determining such schedules. Moreover, most of the researches believe job processing to be deterministic. Some of the researches consider uncertain processing times. Job operations may have a random duration with a probability distribution.

This paper introduces a penalty minimization JSS model under uncertainty. This model is developed to minimize the cost of untimed job penalty and the non-utilized capacity cost. The model considers the job times to be random with a certain probability distribution. The aim is to determine the optimal start times of each task to minimize the expected total penalties. The incurred costs in this model are the delay penalties expressing the tardiness, storage cost representing the earliness, and the non-utilized capacity cost.

Problem Description and Assumptions

Besides the growing trend towards the application of GA to JSSP, researchers are looking for using simple software to aid them to obtain good solutions in a low time. The Microsoft Excel spreadsheet and an add-in to provide the GA called @RiskOptimizer are used in the domain of scheduling problems.

The authors decided to get benefits from this advantage of using @RiskOptimizer. This research uses a Microsoft Excel spreadsheet-based commercial genetic algorithm Evolver with solver @RiskOptimizer to build the JSS model.

The Following Assumptions are Considered in the Model

- The processing times are uncertain;
- Each job has its own due date;
- Each job will visit the same machine not more than one time;
- All jobs and machines are ready at time zero;
- Each machine can process only one job at a time;
- Recirculation is not allowed;

Notation and Model Formulation

Sets

J: Set of jobs

M: Set of machines

Parameters

P_{ji}: Processing time for job j on m/c i

PM_{ji}: The mean value of the processing time for job j on m/c i

PS_{ji}: The standard deviation of the processing time for job j on m/c i

D_j: Due date of job j, j =1, 2, …, J

SC_j: The storage expenses of job j per unit time

DC_j: The penalty of job j delay per unit time

PC_j: The penalty of job j delay

MIP_i: The penalty of machine i idling per unit time

SEQ: Processing sequence array

NUMT: Number of machines (tasks) for each job

NUMJ: Number of jobs per machine J

DISJ: Disjunction array

Decision Variables

S_{ji}: Starting time of job j on machine i,

Fji: Finishing time of job j on machine i,

Cj: Completion time of job j,

Ej: Earliness of job j = (Dj - Cj) if Dj>Cj, and 0 otherwise,

Tj: Tardiness of job j = (Cj - Dj) if Cj>Dj, and 0 otherwise,

$\text{MITi: Idle time of machine i} = \text{MAX(Fji)} - \sum_{j \in N} \text{Pji}$

SPj: Single penaltyfornotaccomplishingjobjontime(tobepaidonce)

TPj: Tardiness penalty of job j.

EPj: Earliness penalty of job j.

TJj: Binary variable = 1 if Tj> 0, and 0 otherwise,

Objective Functions

Total penalty = Non-utilised capacity penalty + Delay penalty (Tardiness) + Single payment penalty (Tardiness) + Storage penalty (Earliness).

The objective of total penalties is given in Equation 1.

$$\text{Total Penalty} = \sum_{i \in M}(\text{MIT}_i * \text{MIP}_i) + \sum_{j \in N}(\text{T}_j * \text{TP}_j) + \sum_{j \in N}(\text{TJ}_j * \text{SP}_j) + \sum_{j \in N}(\text{E}_j * \text{W}_j * \text{EP}_j) \quad (1)$$

Constraints

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

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

Conjunction Constraints

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

Computational Results and Analysis

In this section, the results of applying the proposed model are introduced and analyzed. The model has been solved using @RiskOptimizer solver.

The model accuracy and capability are verified through solving a numerical problem. The processing sequences of three jobs on four machines are shown in table 1. The duration matrix and the due date of each job are shown in table 2 and 3 respectively.

Table 1: Job Sequence

J1	**3**	**2**	**4**	**1**
J2	3	2	1	4
J3	1	3	4	2

Table 2: Duration Matrix

M/J	J1	J2	J3
M1	80	100	50
M2	40	90	25
M3	60	50	80
M4	60	50	50

Table 3: Due Date

Job	1	2	3
Due Date	300	350	320

The optimization process starts with an accurate problem modelling. For any given set of decision values, called adjustable cell values, the model evaluates an objective function, which required to be optimized. @RiskOptimizer searches for the solution, the objective function provides feedback, telling how good or bad solution is. @RiskOptimizercontinues to search for better solutions until no considerable improvements can be obtained in a predefined number of trials.

The problem is solved twice with the objective of minimizing the total penalty. In the first, considering deterministic duration processing times after that the durations are assumed to follow normal distribution N (μ, σ) with mean μ and standard deviation σ. The ratio between the standard deviation and mean is known as variability.

Case 1: Output of the Model with Deterministic Duration

Table 4 to Table 7 illustrate the results of the model with deterministic processing durations. The unit idle penalties are assumed to be 50, 5, 5 and 5 $/time unit for each machine respectively. The resulted in total penalty due to the shortage, delay and a single penalty is equal to 6000 $ while the penalties due to idle time or non-utilized capacity of machines are 6275$. Consequently, the objective function of the total penalty is equal to their sums 12275$**.**

Table 4: The Start and Finish Times of Each Operation

OperationNo.	1	2	3	4	5	6	7	8	9	10	11	12
Job ID	1	1	1	1	2	2	2	2	3	3	3	3
Machine Required	3	2	4	1	3	2	1	4	1	3	4	2
Duration Time	60	40	60	80	50	90	100	50	50	80	50	25
Start Time	50	140	180	240	0	50	140	300	19	110	245	295
Finish Time	110	180	240	320	50	140	240	350	69	190	295	320

Table 5: Due Date, Finish, Earliness and Tardiness

	J1	J2	J3
Due Date	300	350	320
Finish Time	320	350	320
Earliness	0	0	0
Tardiness	20	0	0

Table 6: Earliness, Tardiness and Single Payment Penalties

Penalty	J1	J2	J3	J4	J5	∑
Earliness Penalty per Unit of Time	2	2	1	0	0	
Earliness Penalty (Storage Cost)	0	0	0	0	0	0
Tardiness Penalty per Unit of Time	200	200	120	0	0	
Tardiness Penalty (Delay Cost)	4000	0	0	0	0	4000
Single Penalty for Each Delay	2000	2000	1200	0	0	
No. of Delays	1	0	0	0	0	
Single Penalties	2000	0	0	0	0	2000
Summation						**6000**

Table 7: Idle Time Penalty

Duration Matrix						Sum	Max. Finishing Time	Idle Time	Unit Idle Penalty	Total Penalty
M/J	**J1**	**J2**	**J3**	**J4**	**J5**					
M1	80	100	50			230	320	90	50	4500
M2	40	90	25			155	320	165	5	825
M3	60	50	80			190	190	0	5	0
M4	60	50	50			160	350	190	5	950
Summation										**6275**

Case 2: Output of the Model with Stochastic Duration

This case studies the effect of changing the duration of operations into robust on both the makespan and total penalty. As mentioned before, the duration of operations is assumed to follow the normal distribution. Figures (1-6) show the duration distribution of the operations with (mean time, Standard deviation) of (40, 4), (50, 5), (60, 6), (80, 8), (90, 9) and (100,10). It is noticed that the variability, in this case, is equal to 10%.

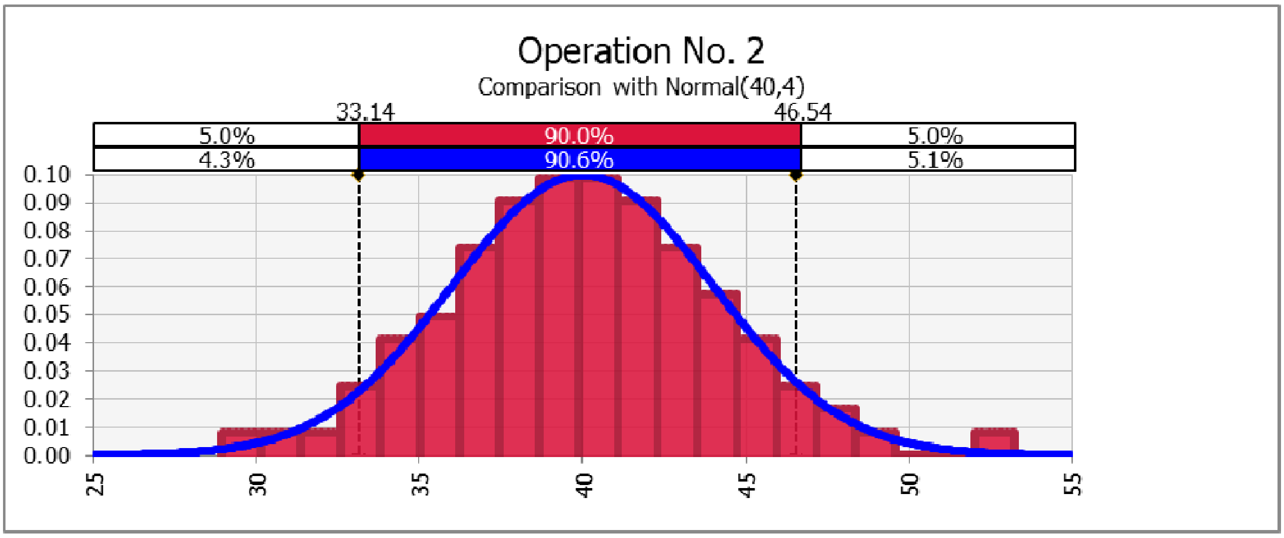

Figure 1: Duration Distribution of Operation No. 2

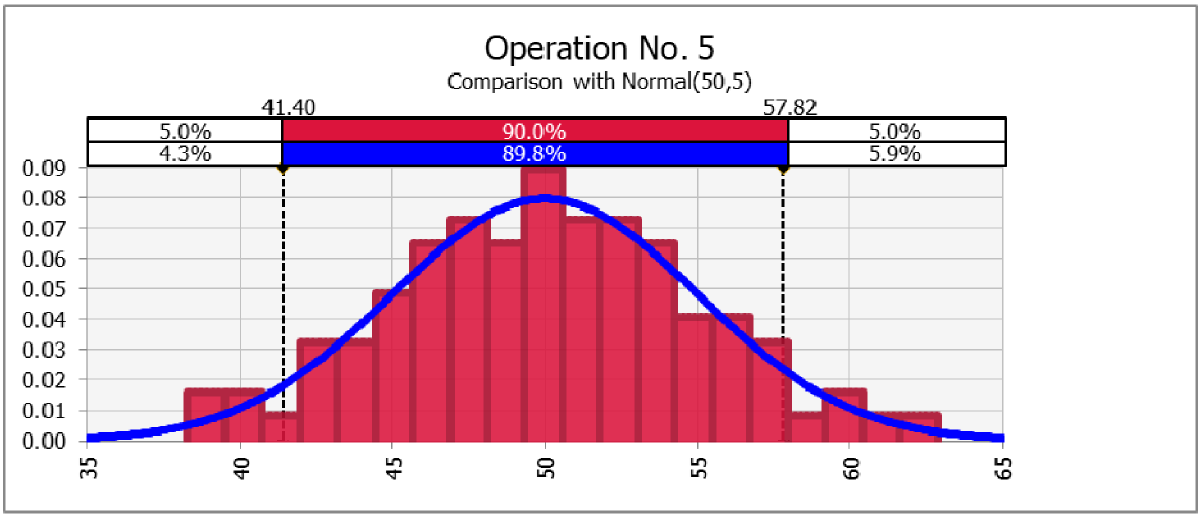

Figure 2: Duration Distribution of Operation No. 5

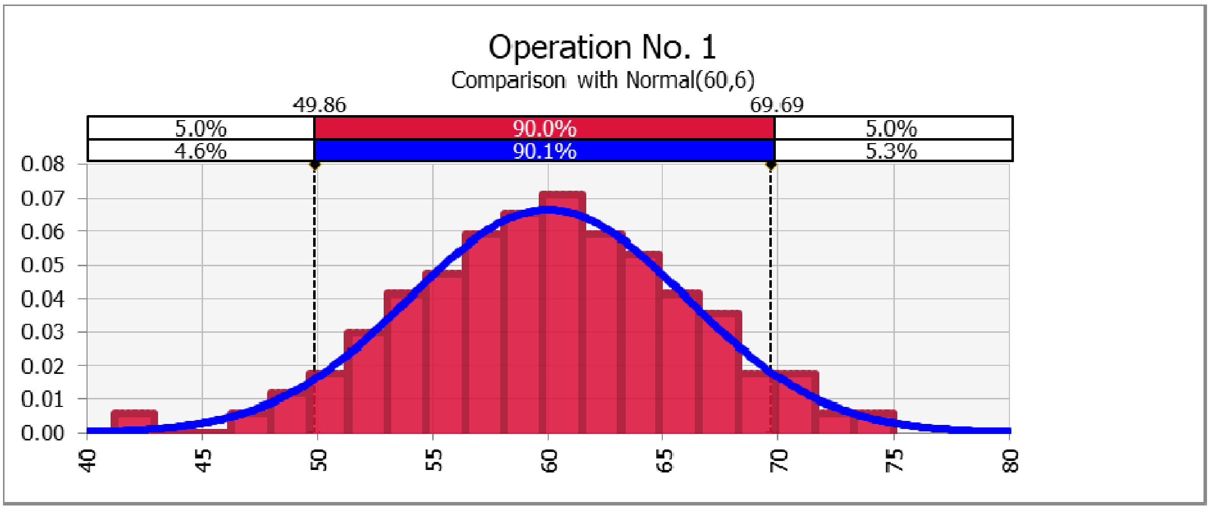

Figure 3: Duration Distribution of Operation No. 1

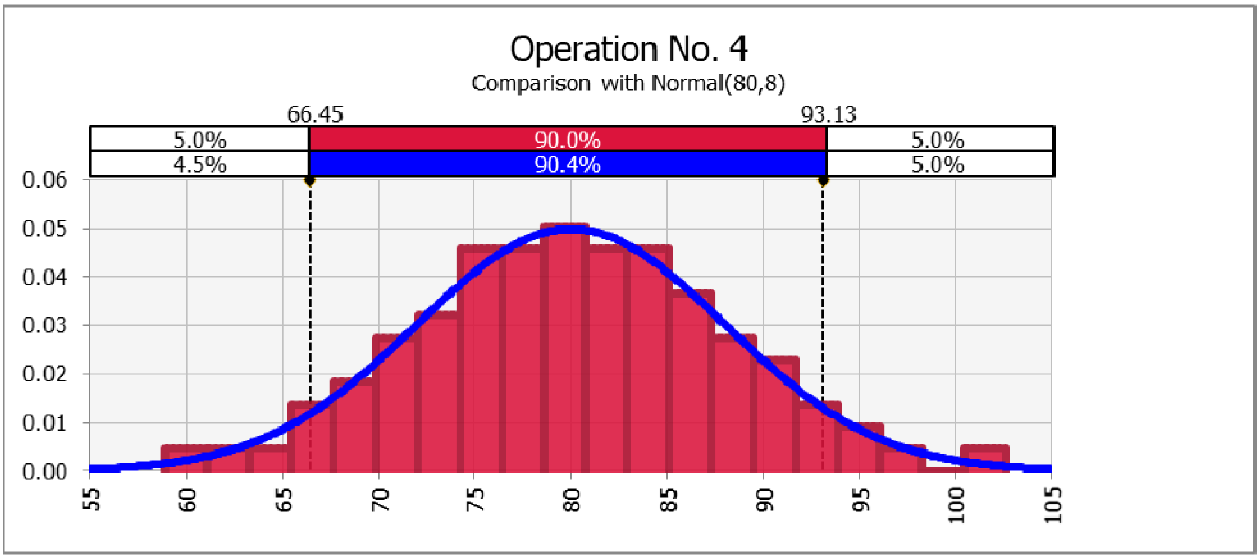

Figure 4: Duration Distribution of Operation No. 4

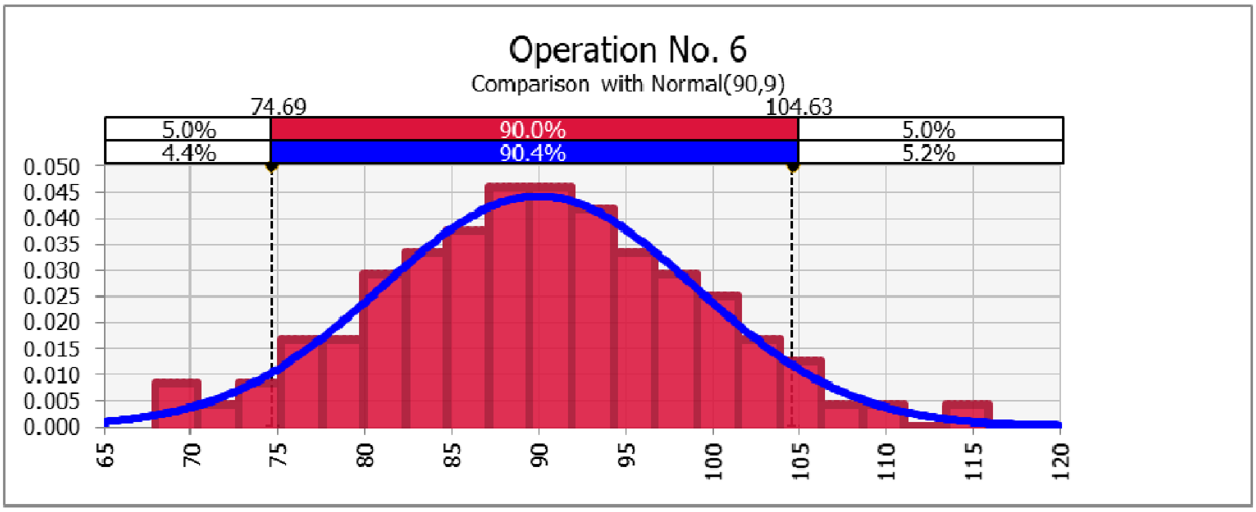

Figure 5: Duration Distribution of Operation No. 6

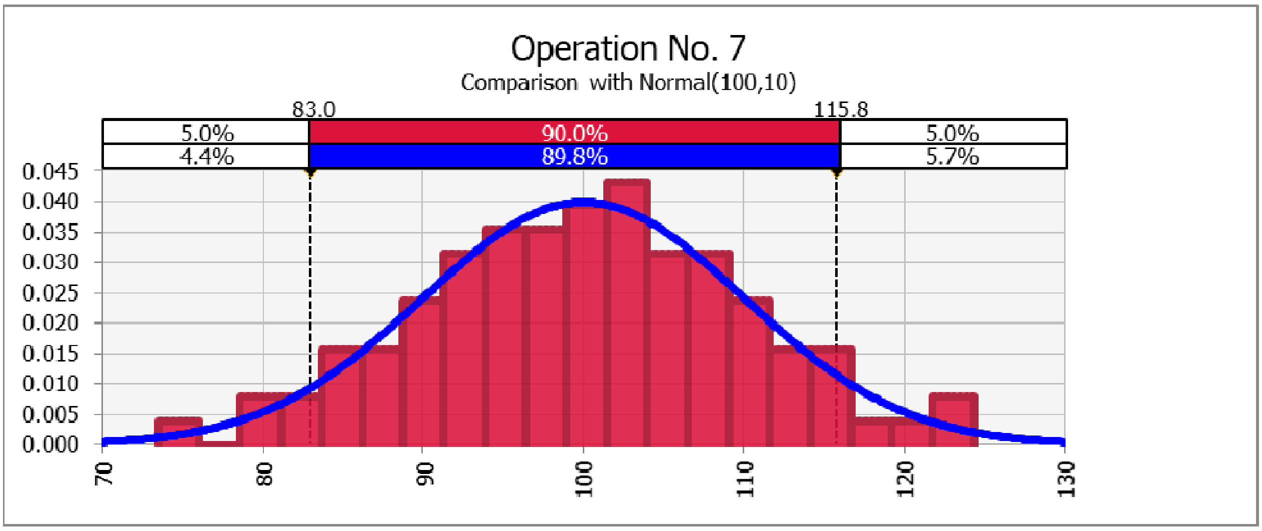

Figure 6: Duration Distribution of Operation No. 7

Figure 7 shows the resulting makespan optimal distribution due to 10% variability. While Figure 8 shows the inputs ranked by the effect on the output. It is clear from Figure 7 that Operation no. 4 has the greatest effect on the makespan. This is more clearly as well from figures (9-11). Operation no.4 has a regression coefficient of 0.7 and 59.115% contribution to variance. It is noticed that Operation No. 8 comes on the second rank to Operation no. 4

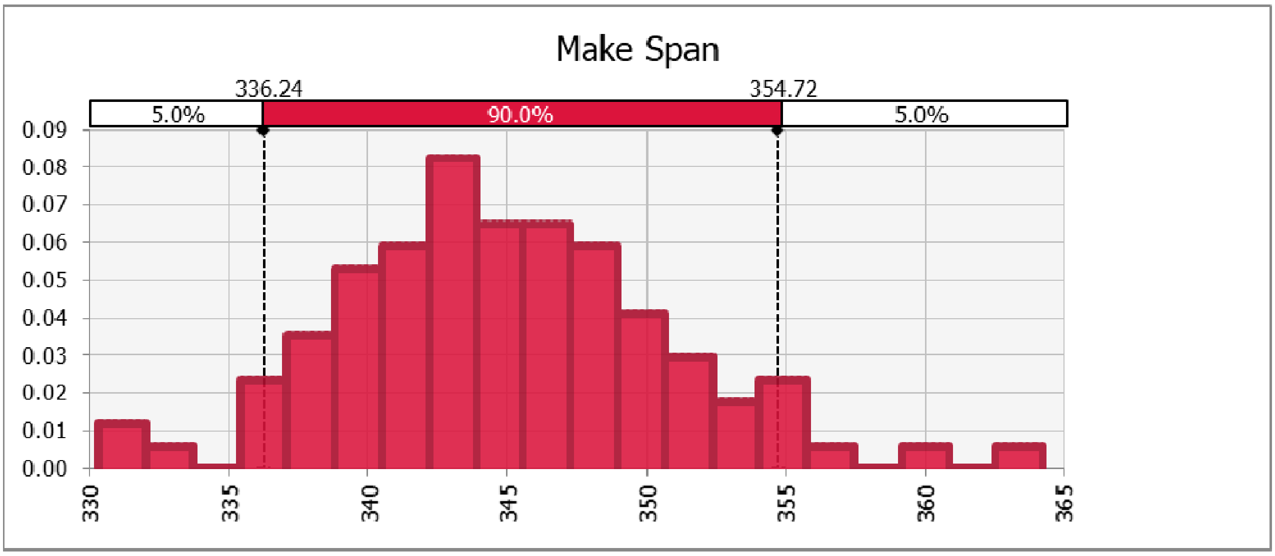

Figure 7: Make span Optimal Distribution Due to 10% Variability

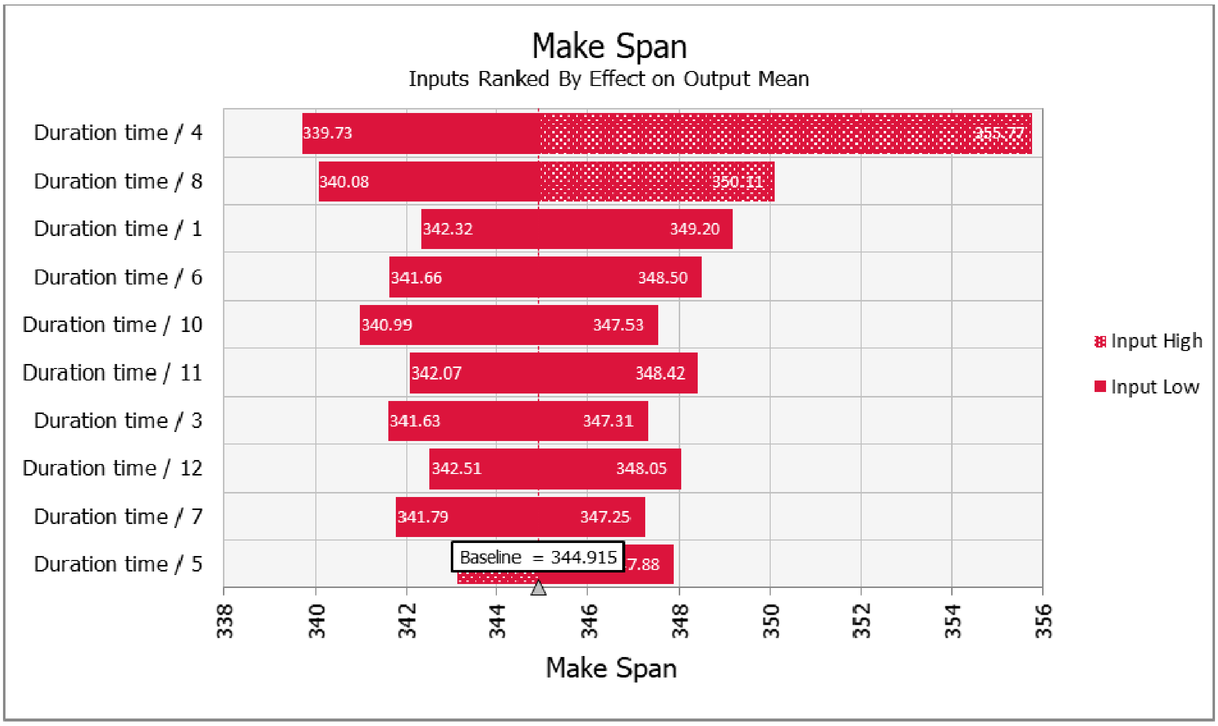

Figure 8: Correlation Coefficients (Spearman Rank)

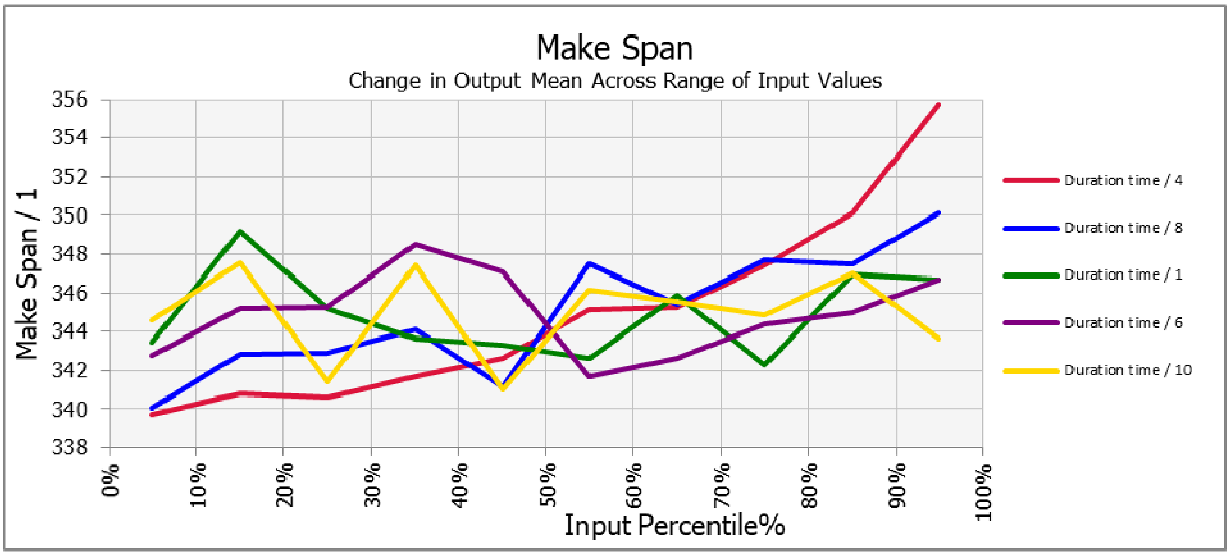

Figure 9: Change in Make span Mean Across Range of Input Values

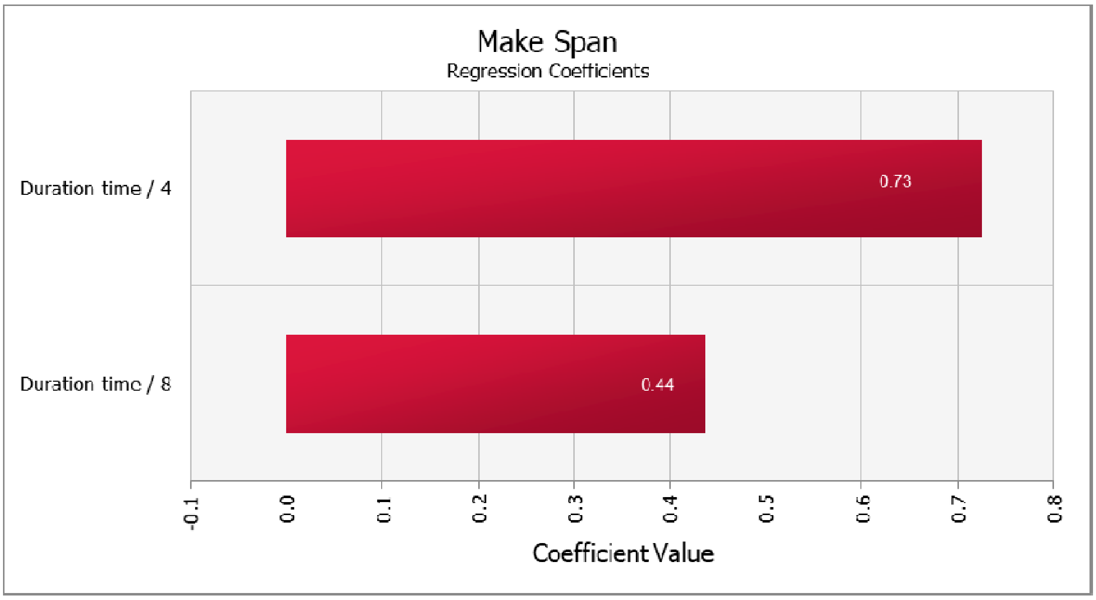

Figure 10: Make span Regression Coefficients

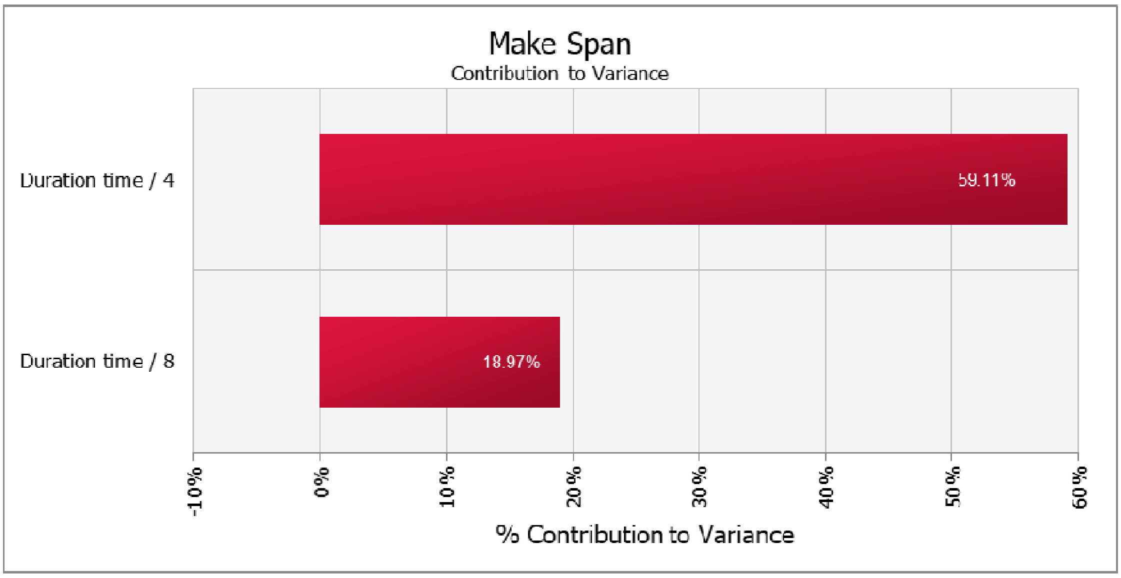

Figure 11: Percentage Contribution to Make Span Variance

Figure (12) shows the resulting total penalties of optimal distribution due to 10% variability. While Figure (13) shows the inputs ranked by the effect on output mean. It is clear from the figure that Operation no.4 still has the greatest effect on the makespan. This is more clearly as well from figures from (14) to (16). Operation no.4 has a regression coefficient of 0.9 and 87.61% contribution to variance. It is noticed that Operation no. 12 come after it in the second rank with a great deviation while Operation no. 8 has a slighter effect.

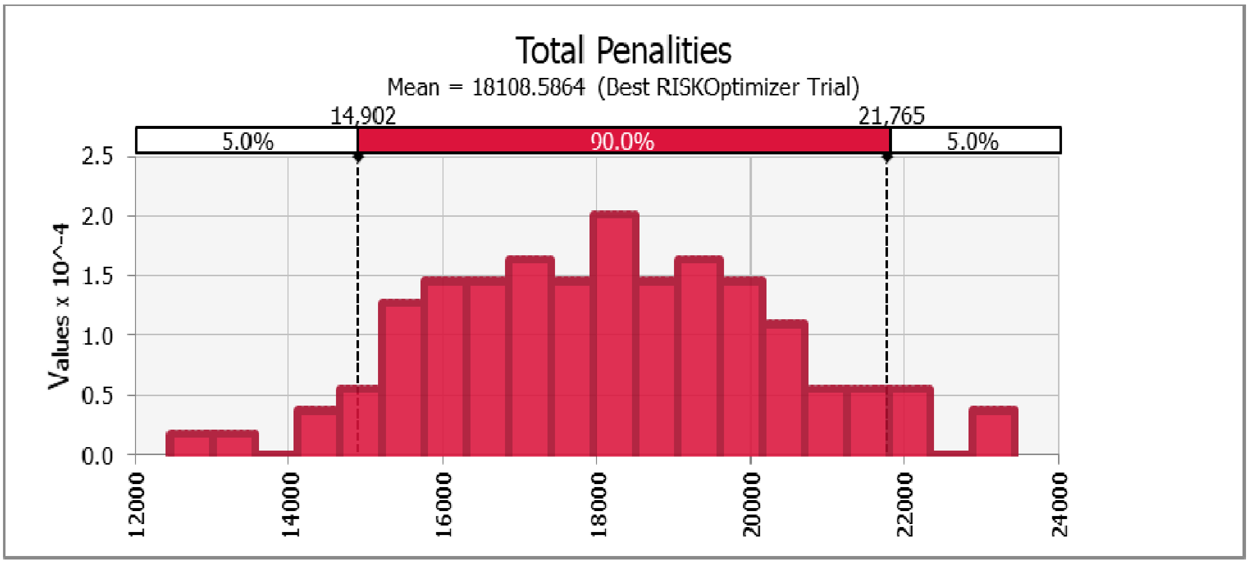

Figure 12: Total Penalties Optimal Distribution due to 10% Variability

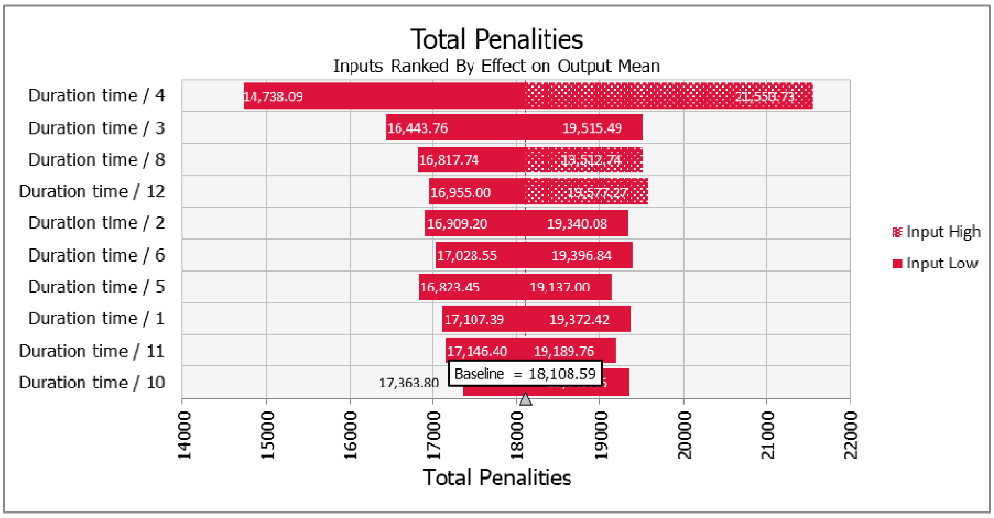

Figure 13: Correlation Coefficients (Spearman Rank)

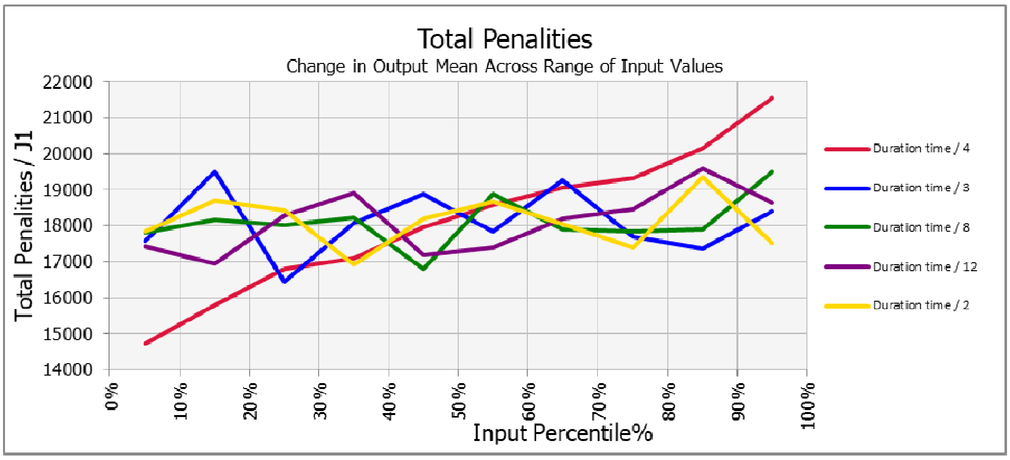

Figure 14: Change in Total Penalties Mean across the Range of Input Values

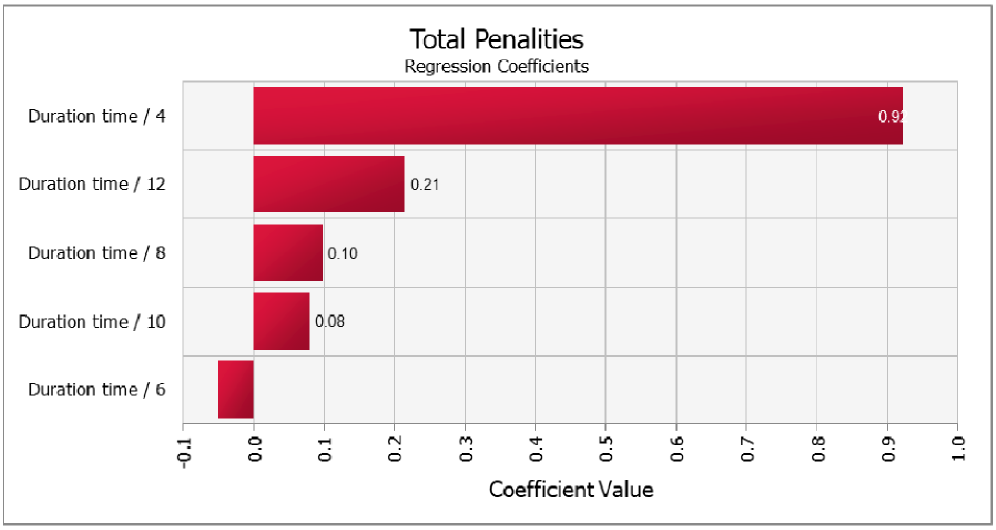

Figure 15: Total Penalties Regression Coefficients

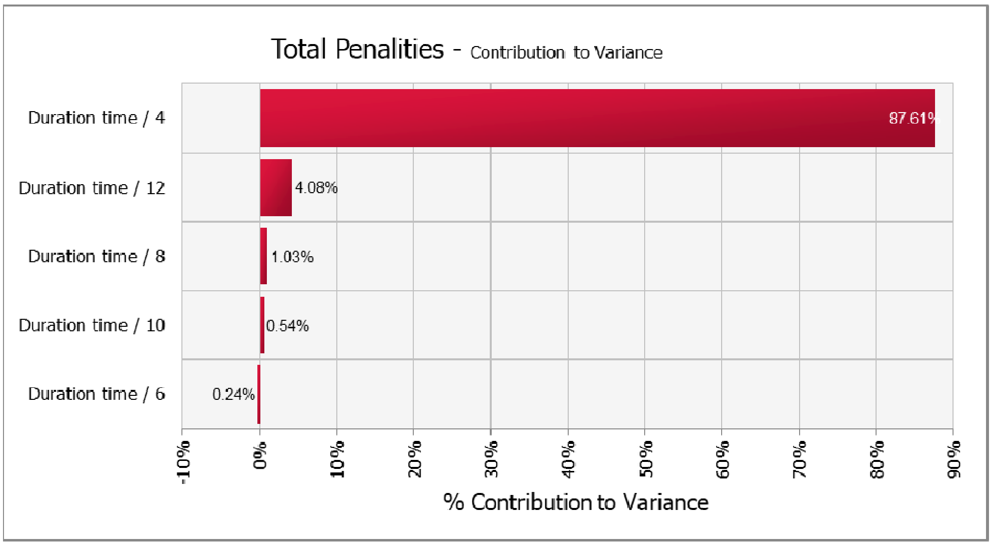

Figure 16: Percentage Contribution to Total Penalties Variance

Table 8summarizes the results of the effect of duration distributions times on the different objective functions.

Table 8: Effect of Duration Normal Distribution on Both Different Functions

	Objective Function			
	Total Penalties		Make Span	
Operation No.	**Regression Coefficient**	**Contribution Variance %**	**Regression Coefficient**	**Contribution Variance %**
4	0.9	87.61	0.7	59.119
12	0.21	4.08	-	-
8	0.10	1.03	0.44	18.97
10	0.08	0.54	-	-
6	-0.05	-0.24	-	-

It is noticed that the effect of changing the duration of operations to follow normal distribution differently affected the objective functions makespan and total penalty. Operation no.4 still have a high effect on both the makespan and total penalties.

Table 9: Results of the Penalty of the Non-Utilized Capacity Cost

M/J	J1	J2	J3	Sum	Idle Time	Unit Idle Penalty	Penalty
M1	80	100	50	230	108.4	50	5418.9
M2	40	90	25	155	161.2	5	805.8
M3	60	50	80	190	14.1	5	70.7
M4	60	50	50	160	187.812341	5	939.0
Total Penalty							**7234**

Table 10: Penalties of Earliness, Tardiness and Single Penalty

Penalty	J1	J2	J3	Sum
Earliness Penalty per unit time	2	2	1	
Earliness Penalty (Storage cost)	0	4.4	3.8	8.2
Tardiness Penalty per unit time	200	200	120	
Tardiness Penalty (Delay cost)	7675.5	0	0	7675.5
Single Penalty for each delay	2000	2000	1200	
No. of delays	1	0	0	
Single Penalties	2000	0	0	2000
Total				**9683.7**

Table 9 represents the results of the penalty of the non-utilized capacity cost while Table 10 represents the penalties of earliness, tardiness, and a single penalty. The comparison between the resultant values of both the deterministic and robust cases is shown in Table 11.

Table 11: Comparison between Deterministic and Robust Results

	Case 1	Case 2
Makespan	350	347.8
The penalty due to idle time	6275	7234.4
Earliness, tardiness and single penalty	6000	9683.7
Sum of Total Penalties	**12275**	**16918.1**

It is clear from the table that the sum of total penalties increases considering uncertain durations. Besides both the idle time, total sum of earliness, tardiness and single penalties increases relative to the case of deterministic duration. The makespan is slightly affected.

CONCLUSIONS

The JSSP has been successfully solved to optimize the total penalty under the situation of the uncertainty of the processing times of all jobs. A comparison was done between the obtained results in different cases. Dealing with the problem of stochastic processing times is more practical. The penalties are important to be calculated besides the makespan.

A developed JSS optimization model has been built using the Microsoft Excel spreadsheets and solved using @Risk solver.

The model has been verified to be accurate and effective through the analysis of the obtained results.

The following are some recommendations that can be considered for future research work:

Researchers can tackle JSSP in multi-period instead of considering a single period only

- Most problems of JSSP were addressed in the static environment. Efforts can be directed to solve a dynamic environment.
- Studying the effect of material selection and design on the obtained schedule and showing how it can affect both operation times and the machine selection which in return affects both the sequence of operations and consequently the schedule.
- Efforts can be devoted to solving the problem under various configurations such as common cycle scheduling problem CCSP and cyclic scheduling problem CSP
- Unexpected disruptions can be studied to show their effects on the schedule. This gives the chance to handle practical cases in the industry.
- Rescheduling is a rich field to do more researches in the future.

REFERENCES

1. *B. Çaliş and S. Bulkan, "A research survey: review of AI solution strategies of job shop scheduling problem," Journal of Intelligent Manufacturing, vol. 26, pp. 961-973, 2015.*
2. *S. Singh and K. P. Singh, "Cuckoo search optimization for job shop scheduling problem," in Proceedings of Fourth International Conference on Soft Computing for Problem Solving, 2015, pp. 99-111.*
3. *B. Wang, X. Wang, F. Lan, and Q. Pan, "A hybrid local-search algorithm for robust job-shop scheduling under scenarios," Applied Soft Computing, vol. 62, pp. 259-271, 2018.*
4. *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.*
5. *M. Faccio, J. Ries, and N. Saggioro, "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.*
6. *Y. Wang, "A new hybrid genetic algorithm for job shop scheduling problem," Computers & Operations Research, vol. 39, pp. 2291-2299, 2012.*
7. *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.*
8. *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.*
9. *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.*
10. *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.*
11. *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.*
12. *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.*
13. *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.*
14. *K. R. Baker and B. Keller, "Solving the single-machine sequencing problem using integer programming," Computers & Industrial Engineering, vol. 59, pp. 730-735, 2010.*
15. *M. Al-Ashhab, "Multi-Objective Job Shop Scheduling Using a Lexicographic Procedure," International Journal of Engineering Science Invent ion (IJESI) vol. 7, p. 10, January 2018.*

16. *R. Sarker, D. Essam, S. K. Hasan, and A. M. Karim, "Managing risk in production scheduling under uncertain disruption," AI EDAM, vol. 30, pp. 289-299, 2016.*
17. *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.*
18. *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.*
19. *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.*
20. *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.*
21. *S. Goren and I. Sabuncuoglu, "Robustness and stability measures for scheduling: single-machine environment," IIE Transactions, vol. 40, pp. 66-83, 2008.*
22. *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.*
23. *F. T. Chan, T. Wong, and L. Chan, "The application of genetic algorithms to lot streaming in a job-shop scheduling problem," International Journal of Production Research, vol. 47, pp. 3387-3412, 2009.*
24. *F. Chan, T. Wong, and L. Chan, "Lot streaming for product assembly in job shop environment," Robotics and Computer-Integrated Manufacturing, vol. 24, pp. 321-331, 2008.*
25. *R.-H. Huang, "Multi-objective job-shop scheduling with lot-splitting production," International Journal of Production Economics, vol. 124, pp. 206-213, 2010.*
26. *P. A. Varthanan, N. Murugan, and G. M. Kumar, "A simulation based heuristic discrete particle swarm algorithm for generating integrated production–distribution plan," Applied Soft Computing, vol. 12, pp. 3034-3050, 2012.*
27. *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.*