International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:6, No:4, 2012

Solving the Flexible Job Shop Scheduling Problem with Uniform Processing Time Uncertainty

Nasr Al-Hinai, Tarek Y. ElMekkawy

***Abstract*—**The performance of schedules released to a shop floor may greatly be affected by unexpected disruptions. Thus, this paper considers the flexible job shop scheduling problem when processing times of some operations are represented by a uniform distribution with given lower and upper bounds. The objective is to find a predictive schedule that can deal with this uncertainty. The paper compares two genetic approaches to obtain predictive schedule. To determine the performance of the predictive schedules obtained by both approaches, an experimental study is conducted on a number of benchmark problems.

***Keywords*—**Genetic algorithm, met-heuristic, robust scheduling, uncertainty of processing times

I. Introduction

FLEXIBLE job shop scheduling problem (FJSP) is computationally difficult problem to solve. However, in real manufacturing systems unforeseen incidents happen. For this, classical models that assume deterministic data about processing times of operations, machines availability; etc; may; in theory; produce an optimal or near optimal schedule, but its performance may deteriorate when implemented in practice; i.e.; released to the shop floor; due to unexpected disruptions. Nevertheless, when incorporating the data uncertainty in the formulation of the already NP-hard FJSP, the problem becomes even more difficult and complicated to solve. A number of methods are suggested in literature to deal with stochastic parameters of a certain scheduling problem. However, based on the desire of the decision maker these methods can be classified and accordingly choose a method that fulfils his need. For example, some decision makers favor a solution that can hedge against the worst possible scenario; others prefer a solution that has a high quality on average; whereas some look for a solution that minimizes the risk of ending with a bad solution. Reference [1] presented a GA that uses sampling technique to estimate the robustness of a single machine schedule subjected to small variation in release dates. They stated that, in a similar way, other types of stochastic problem data can be easily incorporated. In this paper we modified a hybridized genetic algorithm (hGA) proposed by [2] to deal with FJSP when some operations are represented by or subjected to variations characterized by a uniform processing time. The study compares two methods, a method based on sampling technique similar to [1] and a method that optimizes the objective function based on the expected processing time of the operations (i.e. simple method similar to deterministic approach).

Nasr Al-Hinai is with Department of Mechanical & Industrial Engineering, Sultan Qaboos University, P.O. Box 33, P.C. 123 Al-khoud, Oman (phone: 968-2414-2675; fax: 968-2414-1316; e-mail: nhinai@squ.edu.om).

Tarek Y. ElMekkawy is with the Department of Mechanical & Manufacturing Engineering, University of Manitoba, Winnipeg, MB, R3T 5V6, Canada, (e-mail: tmekkawy@cc.umanitoba.ca).

The remainder of this paper is structured as follows: after the literature review in Section II, Section III describes the FJSP. Section IV discusses the modified hGA architecture. Analysis of the computational results is presented and discussed in Section V. Finally, the research summary is covered in Section VI.

II. Literature Review

For decades the emphasis of literature that discusses scheduling problems is put towards deterministic scheduling problems where the data parameters are assumed to be fixed and known beforehand. Nevertheless, recently more attention is given to schedule systems where some data parameters are unknown or are represented by some probabilistic distributions. Since most scheduling problems are classified as NP-hard, heuristic and meta-heuristic approaches received much attention to deal with the presence of uncertainty in the problem's data parameters. This section gives a brief survey of stochastic scheduling approaches found in literature.

References [3]-[9] addressed stochastic single machine with uncertain jobs processing times. Single machine environment subjected to machine breakdowns was considered by others like [10]-[12]. Similarly, [1] used a modified GA to find robust solution in single machine environment subjected to stochastic release dates of jobs.

Also, [13] analysed effects of machine breakdowns and processing time variability on the quality of job shop schedules using slack-time based robustness measure. The performance of simple dispatching heuristics versus algorithmic solution techniques in job shops subjected to uncertain processing times were studied by [14] and [15] showed that dispatching rules are more robust to interruptions than the optimum seeking off-line scheduling algorithms. Reference [16] proposed a two step algorithm based on disjunctive graph representation to minimize maximum lateness and absorb the impact of random machine breakdowns on the predictive schedule of a job shop by inserting idle time. Furthermore, [17] and [18] used GA (proposed in [19]) to improve the robustness and flexibility of the job shop schedules when minimizing maximum tardiness, summed tardiness and total flow-time measures using two robustness measures, a neighbourhood-based robustness measure and a lateness-based robustness measure.

Authors in [20] presented a fuzzy mathematical model of scheduling parallel machines with sequence-dependent cost while considering uncertainties in processing times. [21] proposed a two-stage scheduling decision framework to execute schedules of a two-machine flow shop with interval processing times. Also, [22] proposed a probabilistic generalization to design robust *a priori* scheduling that assumes the number of jobs to be processed on parallel machines as a random variable with respect to the total weighted flow time. Furthermore, [23] presented a real-time simulation-based decision support system to control the production of a stochastic flexible job shop subjected to stochastic processing times. Readers are referred to [24]-[27] who gave detailed review of literature related to scheduling under uncertainty.

In light of the literature, scheduling under uncertainty can be classified into number of categories depending on the adopted strategy by the decision maker on how to react to uncertainties. Hence, methods compared in this paper falls under the category *proactive (robust) scheduling* which is defined as a schedule with relatively insensitive quality to a changing environment ([25] and [26]). Furthermore, the choice of the optimization method depends on the level of uncertainty. Therefore; in our opinion; if the expected level of uncertainty is low enough decision makers and/or schedulers might consider using two possible optimization-based algorithms. The first is to adopt an algorithm that optimizes a schedule based on the average values of parameters with uncertainties (such as processing time). The second is to implement an algorithm based on sampling technique from the random distributions of these parameters. The later approach subjects different schedules' sequences to different sets of uncertainties and then selects the one that performs well on average. To the best of our knowledge, there is not a previous study that addresses a comparison of the two former algorithms for obtaining predictive schedules of the FJSP when some operations are represented by or subjected to low-to-medium processing time variations. Hence, the goal of this work is to evaluate and compare the quality and the solution robustness of predictive schedules obtained using these two choices in flexible job shop environment where the processing times of some operations are represented by or subjected to low-to-medium uncertainty. Specifically, processing times of these operations are represented by an interval of equally possible real value between given lower and upper bounds. For clarity and ease of referencing, the algorithm that optimizes expected average data will be referred to as MS_{exp} and the algorithm that is based on sampling will be be referred to as $MS_{\%\text{Rob.}}$

III. Problem Description

This work considers a non-preemptive flexible job shop scheduling problem (FJSP) with the objective of minimizing the makespan. There is a set of $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ jobs and each job i has a set of $O = \{O_{i1}, O_{i2}, \ldots, O_{iq_i}\}$ operations where q_i denotes the total number of operations of job J_i. Each operation O_{ij} is to be processed in a subset of machines $\mathcal{M}_{ij} \in \mathcal{M} = \{M_1, M_2, \ldots, M_m\}$. An operation O_{ij} cannot start processing until its precedence operation $O_{i(j-1)}$ has finished its processing. All n jobs are available at time $t = 0$ and the processing time p_{ijk} of some operations O_{ij} of job J_i in machine M_k may equally take any real value between given lower $\underline{p}_{ijk}$ and upper $\bar{p}_{ijk}$ bounds. This processing time variation of operation is due to, e.g., incomplete or unreliable information or unavoidable stochastic variability related to machine's tools and/or workers skills, etc. The processing time uncertainty can be described by a set of all possible scenarios (infinite) ζ. Each unique set of processing times ξ is obtained by equally selecting a value from the associated interval of each operation:

$$p_{ijk}^{\xi} \in \left[\underline{p}_{ijk}, \bar{p}_{ijk}\right] \forall\, i \in \{1, 2, \ldots, n\}, \quad j \in \{1, 2, \ldots, q_i\}, k \in \{1, 2, \ldots, m\} \tag{1}$$

In practice the actual operations' processing time of some operations may not be known or difficult to verify until the operation has finished processing. In such case, assuming an expected value helps the scheduler or decision maker in obtaining a schedule that satisfies a certain performance measure. In this work, the expected processing times $E\left[p_{ijk}\right]$ of operations that are represented by or subjected to variations according to a uniform time intervals are given by:

$$E\left[p_{ijk}\right] = \left(\left(\underline{p}_{ijk} + \bar{p}_{ijk}\right)/2\right) \tag{2}$$

IV. Hybridized Genetic Algorithm for The FJSP

Reference [2] proposed hybridized genetic algorithm (hGA) architecture for the deterministic FJSP and results illustrated that the approach is very effective in minimizing the makespan of this problem. Recently, authors ([28]) showed that this hGA can be modified to deal with FJSP subjected to random machine breakdowns by replacing its fitness function. In the following subsections, we describe the original deterministic hGA and then show how it can be modified to minimize the makespan of schedules according to average expected processing times data or according to the sampling technique method.

A. Deterministic hGA for the FJSP

Reference [2] used permutation-based representation chromosome representation; where each operation is represented by triples (k,i,j) such that k is machine assigned to the operation, i is current job number, and j is the progressive number of that operation within job i. A schedule for FJSP with three jobs and three machines can be represented by (221-131-111-212-322-223-332). In this architecture, the initial population is created by two ways. The first way is to generate half of the population randomly. The second half of the population is generated using a schedule construction heuristic called *Ini-PopGen*. Ini-PopGen starts by randomly assigning priority to jobs. Then, based on this priority an operation is scheduled on the machine (from the set of appropriate machines) that can finish it sooner. This procedure considers the processing time and the work load on the machine while assigning operations.

Chromosomes decoding follow an active decoding procedure, wherein no operation can be started earlier without delaying at least one other operation or violating the technological constraints. After the active decoding, the schedule is improved by a local search procedure that results in a local optimal schedule (Lamarckian learning). However, this local search procedure is only applied every d^{th} generation and number of moves is limited to a maximum *loc_iter* moves without improvements.

Two chromosomes are selected from the population. At first, roulette wheel technique is used to form donors' mating-pool based on a selection probability given by:

$$P_{sel} = \frac{F_{ind}}{F_{tot}}, ind = 1, \dots, N \quad (3)$$

Where, P_{sel}: is the probability of choosing the ind^{th} individual; N: is the population size; F_{ind}: is the ind^{th} individual fitness; and F_{tot}: is the total fitness of all individuals in the current generation.

Then, if the individual in the donors' mating-pool passes a crossover probability P_c, an n-Size tournament method is used to select n chromosomes from the population to form the receivers' mating-sub-pool. Then, the best individual (one with lowest fitness value, makespan) in the sub-pool is chosen for reproduction.

The crossover operator is based on the *Precedence Preserving Order-based Crossover* (POX) ([29]) and was modified not to treat the parents symmetrically. Mutation of individuals is implemented through using two operators. The first operator is a *Machine Based Mutation* (MBM), where a random number of operations (denoted as *nrand*) are selected and reassigned to another machine. After that, modified *Position Based Mutation* (PBM) is applied. PBM was originally designed for JSP using single triple permutation-based chromosomes representation. Thus, the PBM is modified so that no infeasible chromosomes are produced and it starts by randomly selecting an operation within the chromosome and then reinserting it at another position.

B. Modified hGA for the FJSP

In this section, we describe how the original deterministic hGA can be modified to solve the FJSP to find schedules using the two methods, MS_{exp} and $MS_{\%\text{Rob}}$.

Historical records of a certain shop floor can provide approximated distribution uncertainties that can affect it; such as machine breakdowns, processing times variations, cancelations or arrivals of new jobs, etc. In FJSP, [28] showed that such distributions can be used as a guide when generating the predictive schedule. This can be achieved by integrating the probability distribution of that specific uncertainty with the machine routing and sequencing of operations so that overall performance, measured by makespan, of the schedule is not affected to a high degree in case such disruption occurs.

Previous studies like [1], [13], [17], [18], [28], etc, showed that such objective can be achieved by replacing the fitness function of the GA by a fitness function that satisfies the new objective, usually referred to as *robust fitness function*. The main purpose of such robust fitness evaluation functions is to guide the evolution of solutions towards solutions that are not or slightly affected by perturbed data parameters.

For the suggested comparison between the two methods in this work, the ordinary objective function of the deterministic FJSP with minimum makespan is given by:

$$MS_{min} = min\,\{max(C_i)\}\forall J_i = \{\mathrm{J}_1, \mathrm{J}_2, \dots, \mathrm{J}_n\} \quad (4)$$

where, MS_{min} is the minimum makespan, and C_i is the completion time of job J_i; can be applied and/or modified as follows. First, the same ordinary objective function (4) is used for optimizing the MS_{exp} method. The only difference is when using the processing times of operations represented by or subjected to uncertainty. In this case, the expected processing operations' times replaces the uniform interval processing times and then these expected processing time values are used to generate the sequence of the predictive schedule. However, for the second method $MS_{\%\text{Rob}}$ the procedure is not straight forward. Here, according to [30], the solution of such objective fitness function has to be implemented on a randomly modified sample set of characteristics (or data parameters) and then combining a number of evaluations of the same schedule *s* sequence solution in the objective fitness function. A possible sampling objective fitness for uncertain processing times can be represented by a weighted average of *m* derived evaluations such that:

$$MS_{\%\text{Rob}}(s) = \frac{1}{m}\sum_{l=1}^{m} w_l \varphi\left(MS_{min}(s), \zeta_l\left(p_{ijk}^{\xi}\right)\right)$$
$$\forall\ i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, q_i\}, k \in \{1, 2, \dots, m\} \quad (5)$$

where, wl is the weight related to the derived schedule s sequence evaluation, $\zeta_l\left(p_{ijk}^{\xi}\right)$ is sampling function that takes a random sample of a certain processing time scenario p_{ijk}^{ξ}, and m is the number of samples used to evaluate the schedule *s*.

Therefore, the previously described hGA in subsection IV-*A* is modified to first use (4) with the expected processing times for MSexp method, and then modified by replacing its fitness function by (5) for $MS_{\%\text{Rob}}$ method. The hGA used for each method will be referred to as MS_{exp}-hGA and $MS_{\%\text{Rob}}$-hGA, respectively. Using different objective fitness function in the hGA will lead to obtaining different schedule sequences. Thus, each schedule's sequence may respond differently to disruptions as some may be able to absorb their effects more than others. Furthermore, since GA utilizes a population of solutions in its search, this gives higher chances to explore a diverse set of solutions. Hence, GA will have higher chances of finding schedule sequences that are less sensitive to data uncertainties.

V. Analysis and Results

Numbers of FJSP benchmarks with a wide range of sizes, from 5 x 3 to 20 x 15 found in literature are used for the experiments. These benchmarks are Ex1 taken from [31], and examples MK01, MK03, MK07 and MK10 proposed by [32]. For this work, the expected processing time $E\left[p_{ijk}\right]$ of an operation is to be equal to processing time of that operation in the original problem. Hence, the upper and lower processing time bounds of an operation affected by uniform variation in its processing time is calculated by:

$$\left[\underline{p}_{ijk}, \bar{p}_{ijk}\right] = E\left[p_{ijk}\right] \times [(1-\beta), (1+\beta)] \quad (6)$$

where, β is the percentage difference from the original expected processing time.

Parameter β represents the level of variability of the operation's processing time. In order to control the number of operations that are affected by the processing time variability, the parameter α has been used. Table I shows different combinations of the two parameters, α and β, that are used to generate the different test cases of the experiments.

TABLE I
DIFFERENT PROCESSING TIME VARIATION'S COMBINATIONS

Disturbance type	% of affected operations & disturbance level	α	β
DT1	Low, low	0.20	0.15
DT2	Low, medium	0.20	0.40
DT3	Medium, low	0.40	0.15
DT4	Medium, medium	0.40	0.40

A. Hybridized genetic algorithm parameters

The number of function evaluations m, i.e. the number of samples used to evaluate the schedule s, in (5) requires being sufficient. This is due to the fact that using a smaller m value may lead to selecting a non-robust solution, whilst using a larger value leads to unacceptable increase in the computational time. Hence, the selected value of m is related to the total number of operations of each instance. The *m value* is set to 50% of the total number of operations, and hence, the corresponding $MS_{\%Rob}$-hGA is referred to as MS_{50Rob}-hGA. All sequence evaluations are given the same importance and hence the weight w_l in (5) is set to 1.

All test codes are implemented and executed using C++ on an Intel® Core™ 2 Quad CPU @ 2.4 GHz with 3.24 GB RAM. For comparability and ease of implementation, all hGA are closely related and the parameters are experimentally tuned according to the performance of MS_{exp}-hGA (minimizing MS_{exp}). The parameter values that are chosen for the two-stage hGA algorithm are as follows: population size 200, crossover probability 0.7, mutation probability 0.3, number of generations 200, number of parents in the receivers' mating sub-pool 4, number of generations to perform local search d = 10, maximum number of moves without improvement in the local search *loc_iter* = *min* [*tot_noper*, 150], and the worst chromosome is replace every k = 3 generations.

B. Analysis of robustness measures

To compare the performance of both methods, MS_{exp}-hGA and $MS_{\%50}$-hGA, a simulation procedure is applied. Consequently, a standard MS_{exp}-hGA using the ordinary evaluation function (described in subsection IV-*B*) minimizing MS_{exp} is first run to obtain schedules with sequence referred to as *expected sequence*. Then, $MS_{\%50}$-hGA, with the systematic application of sampling function evaluation (5), is used to obtain schedule with sequence referred to as *sampling sequence*. In order to draw more accurate responses, five schedules for each of the hGA different settings of each test case is used. After the sequences are obtained, 400 replications of each problem instance with randomly modified processing times according to the disturbances are evaluated. This results in 5 (number of obtained schedules' sequences) x 4 (disturbances levels) x 400 (replications) = 8000 test runs per test instance.

Since this comparative study is done to compare the performance of the predictive schedules' sequence obtained using MS_{exp} method and predictive schedules' sequence obtained using the $MS_{\%Rob}$ method, all obtained sequences are subjected to the same processing time variation disturbances and their performance is compared in terms of:

1) The relative error (*RE*) predictive makespan deviation with respect to the best-known lower bound value defined as:

$$RE = \left[\left(MS_{comp} - LB\right)/LB\right] \times 100 \quad (7)$$

where, MS_{comp} is the initial predicted makespan obtained using either method, and *LB* is the best-known lower bound. It is worth pointing out that since for every replication the processing times are randomly modified, estimating its *LB* value is not possible. Therefore, the used *LB* is the same *LB* reported in literature for the same test case group. The relative error measures the robustness in the objective function space, i.e. quality robustness.

2) The average absolute relative makespan deviation between the initial predicted schedule makespan and the actual realized makespan after the 400 disturbances' replications according to the following equation:

$$Ave.Abs\ RMS_{\Delta} = \frac{1}{N}\sum_{p=1}^{400}\left\{\sqrt{\left[MS(q)_{Rp} - MS(q)_{P}\right]^2/[MS(q)_{P}]^2}\right\} \times 100 \quad (8)$$

where, $\boldsymbol{q}$ is the replication predictive schedule, and the subscripts P, R and p : refer to predictive (or the original released schedule to the shop floor), realized (or the actual schedule after disturbance simulation), and the processing time disturbance number, respectively.

The performance measures addressed above examine the average values related to the obtained replications' schedules at each combination before or after the disruptions. One of the essential concerns associated with any proposed method, a heuristic or a meta-heuristic method, to solve a problem is arbitrating the quality of its obtained solutions (predictive schedules in this study). Therefore, the first measure *RE* seeks to answer that concern by measuring how far are the obtained schedules from the optimal or near optimal schedules? This ensures that only methods that are capable of obtaining predictive schedules of high quality, minimum makespan, as well as proving their repeatability, or robustness in obtaining such schedules, are given the credit. For this reason, RE measure is designed to work on the objective function space by comparing the quality of obtained solution of any method to a standard benchmark solution.

While RE measure quantifies the quality robustness of obtained predictive schedules, and $Ave.Abs\ RMS_{\Delta}$ measures, on the other hand, assist evaluating the solution robustness of these schedules and their sustainability in the face of uncertainties. To achieve this evaluation it is required to find a way of comparing the original solution, i.e. predictive schedule released to the floor shop, to the final solution, i.e. the realized schedule after the disruptions.

This is achieved by using the *edit distance* concept introduced by [33]. This concept is generalized to measure the distance between two schedules by considering the difference in the main performance measure between the predictive schedule and the realized schedule, i.e. makespan deviation. For this reason, $Ave.Abs\ RMS_{\Delta}$ *measure* is used and is interpreted as the relative mean of the deviations between the realized schedules after disruptions and the originally released predictive schedules. Moreover, it is considered as a quantity that measures how close, in average, the realized schedules are to the predictive schedules.

C. Computational results

Table II shows the detailed results obtained using MS_{exp}-hGA and $MS_{\%50}$-hGA. Due to space limitations, Table II is divided into two parts. It consists of 10 columns. The first column represents the instance name and size. The second column refers to used method to obtain the corresponding schedule. The remaining columns are labelled according to the performance measures given above and give the results of the 400 replications of specific disturbance type for the modified instances, i.e. test case groups. For each column, the best performance; lowest average deviation percentage; is printed in bold-face. When considering RE results in Table II, it can be noted that including variations of the processing times in the objective function ($MS_{\%50}$-hGA) to obtain a schedule has a negligible effect on increasing the makespan of the predictive schedule. Therefore, the maximum increase in RE when using the robust sampling objective function, $MS_{\%50}$-hGA, compared to the expected ordinary fitness function evaluation, MS_{exp}-hGA, is 6.09% (for the Ex1 test case group) and on average 0.61% for all modified instances. Furthermore, in most considered cases the sampled solutions obtained by $MS_{\%50}$-hGA were sometimes slightly better than the expected solutions obtained by MS_{exp}-hGA like for the test cases MK01, MK07, and MK10. This may be explained by the change in how the population is handled when using the sampling function evaluation, (5), which may allow escaping from local optima. In addition, in terms of $Ave.Abs\ RMS_{\Delta}$, the robust solutions results acquired by $MS_{\%50}$-hGA are outperforming those obtained by MS_{exp}-hGA. Thus, using a schedule sequence obtained by $MS_{\%50}$-hGA performs mostly better after disturbance occurrence compared to a schedule obtained by MS_{exp}-hGA. These findings highlight the capability of $MS_{\%50}$-hGA to find solutions that are both quality robust and solution robust.

TABLE II
COMPUTATIONAL RESULTS – DEVIATION OF SCHEDULES WHEN SUBJECTED TO RANDOM UNIFORM PROCESSING TIME VARIATIONS

Inst. & Size	Method	DT1 RE	DT1 $Ave.Abs\ MS_{\Delta}$	DT2 RE	DT2 $Ave.Abs\ MS_{\Delta}$
Ex1 (5 x 3)	MS_{exp}-hGA	**52.17**	1.47	**52.17**	4.09
	MS_{50R}-hGA	56.52	**1.37**	58.26	**3.56**
MK01 (10 x 6)	MS_{exp}-hGA	13.89	1.14	13.89	2.77
	MS_{50R}-hGA	13.33	1.16	**11.11**	**2.43**
MK03 (15 x 8)	MS_{exp}-hGA	**0.00**	0.88	**0.00**	**2.29**
	MS_{50R}-hGA	**0.00**	0.90	**0.00**	2.44
MK07 (20 x 5)	MS_{exp}-hGA	10.53	0.83	10.53	2.49
	MS_{50R}-hGA	10.98	**0.80**	**9.32**	2.38
MK10 (20 x 15)	MS_{exp}-hGA	39.64	0.80	39.64	2.34
	MS_{50R}-hGA	39.39	**0.62**	40.36	2.23

TABLE II (CONTINUE)
COMPUTATIONAL RESULTS – DEVIATION OF SCHEDULES WHEN SUBJECTED TO RANDOM UNIFORM PROCESSING TIME VARIATIONS

Inst. & Size	Method	DT3 RE	DT3 $Ave.Abs\ MS_{\Delta}$	DT4 RE	DT4 $Ave.Abs\ MS_{\Delta}$
Ex1 (5 x 3)	MS_{exp}-hGA	**52.17**	1.97	**52.17**	5.48
	MS_{50R}-hGA	53.91	1.91	54.78	4.91
MK01 (10 x 6)	MS_{exp}-hGA	13.89	1.49	**13.89**	3.81
	MS_{50R}-hGA	**12.22**	**1.41**	16.67	**3.68**
MK03 (15 x 8)	MS_{exp}-hGA	**0.00**	1.17	**0.00**	3.24
	MS_{50R}-hGA	**0.00**	1.19	**0.00**	3.12
MK07 (20 x 5)	MS_{exp}-hGA	10.53	1.37	10.53	4.11
	MS_{50R}-hGA	9.92	**1.17**	11.28	3.92
MK10 (20 x 15)	MS_{exp}-hGA	39.64	1.17	**39.64**	3.80
	MS_{50R}-hGA	**38.67**	0.90	40.48	3.41

Values written in bold are the best values

VI. CONCLUSION

This paper presented how a hybridized genetic algorithm for a flexible job shop problem can be modified to find robust solutions when it is subjected to low-to-medium random variations in the operations' processing times. For this, two methods were compared. Our computational results showed that obtained solutions are both solution robust and quality robust. Furthermore, computational results revealed an interesting finding showing if an FJSP is subjected to a low-to-medium level of processing time uncertainty, then an optimization-based method working with expected processing times value may obtain schedules that are as good as schedules obtained using a sampling technique method and hence saving the computational efforts.

As a future research direction, the current research can be extended to study the impact of other kinds of processing time distributions on the conclusions found in this study. Currently, authors are exploring extending this study by developing a multi-objective approach that returns the Pareto frontier solutions so that a decision maker can select a preferable robust and/or stable schedule.

REFERENCES

[1] M. Sevaux, and K. Sörensen, “A genetic algorithm for robust schedules in a one-machine environment with ready times and due dates,” *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 4OR 2, 2004, pp. 129–147.

[2] N. Al-Hinai, and T. ElMekkawy, “An efficient hybridized genetic algorithm architecture for the flexible job-shop scheduling problem,” *Flexible Services and Manufacturing Journal*, vol. 23, 2011, pp. 64-85, doi: 10.1007/s10696-010-9067-y.

[3] R. L. Daniels, and P. Kouvelis, “Robust scheduling to hedge against processing time uncertainty in single-stage production,” *Management Science*, vol. 41, no. 2, 1995, pp. 363-376.

[4] P. Kouvelis, R. L. Daniels, and G. Vairaktarakis, “Robust scheduling of a two-machine flow shop with uncertain processing times,” *IIE Transactions*, vol. 32, 2000, pp. 421-432.

[5] R. H. Möhring, F. J. Radermacher, and G. Weiss, “Stochastic scheduling problems II: Set strategies,” ZOR – *Zeitschrift für Operations Research*, vol. 29, 1985, pp. 65–104.

[6] R. Montemanni, “A mixed integer programming formulation for the total flow time single machine robust scheduling problem with interval data,” *Journal of Mathematical Modelling Algorithms*, vol. 6, 2007, pp. 287-296.

[7] M. Pinedo, “On the computational complexity of stochastic scheduling problems,” in *Deterministic and stochastic scheduling, Dordrecht: Reidel*, M. Dempster, J. Lenstra, and A. R. Kan (Eds.), 1982.

[8] C. W. Wu, K. N. Brown, and J. C. Beck, “Scheduling with uncertain durations: Modeling β-robust scheduling with constraints,” *Computers & Operations Research*, vol. 36, 2009, pp. 2348-2356.

[9] Y. Xia, B. Chen, and J. Yue, “Job sequencing and due date assignment in a single machine shop with uncertain processing times,” *European Journal of Operational Research*, vol. 184, 2008, pp. 63-75.

[10] U. M. Al-Turki, J. Mittenthal, and M. Raghavachari, “The single-machine absolute-deviation early-tardy problem with random completion times,” *Naval Research Logistics*, vol. 43, 1996, pp. 573–587.

[11] X. Cai, and F. S. Tu, “Scheduling jobs with random processing times on a single machine subject to stochastic breakdowns to minimize early-tardy penalties,” *Naval Research Logistics*, vol. 43, 1996, pp. 1127–1146.

[12] L. Liu, H. Gu, and Y. Xi, “Robust and stable scheduling of a single machine with random machine breakdowns,” *International Journal of Advanced Manufacturing Technology*, vol. 31, 2007, pp. 645-654.

[13] V. J. Leon, S. D. Wu, and R. H. Storer, “Robustness measures and robust scheduling for job shops,” *IIE Transactions*, vol. 26, no. 5, 1994, pp. 32-43.

[14] S. R. Lawrence, and E. C. Sewell, “Heuristic, optimal, static, and dynamic schedules when processing times are uncertain,” *Journal of Operations Management*, vol. 15, 1997, pp. 71-82.

[15] I. Sabuncuoglu, and S. Karabuk, “Rescheduling frequency in an FMS with uncertain processing times and unreliable machines,” *Journal of Manufacturing*, vol. 18, no. 4, 1999, pp. 268-283.

[16] S. V. Mehta, and R. M. Uzsoy, “Predictable scheduling of a job shop subjected to breakdowns,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 3, 1998, pp. 365-378.

[17] M. T. Jensen, “Improving Robustness and Flexibility of Tardiness and Total Flow-Time Job Shops Using Robustness Measure,” *Applied Soft Computing*; vol. 1, 2001, pp. 35-52.

[18] M. T. Jensen, “Generating robust and flexible job shop schedules using genetic algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, 2003, pp. 275-288.

[19] D. C. Mattfeld, *Evolutionary Search and the Job Shop*, in *Production and Logistics*, Physica-Verlag, 1996.

[20] A. Anglani, A. Grieco, E. Guerriero, and R. Musmanno, “Robust scheduling of parallel machines with sequence-dependent set-up costs,” *European Journal of Operational Research*, vol. 161, 2005, pp. 704–720.

[21] N. M. Matsveichuk, Yu. N. Sotskov, N. G. Egorova, and T. C. Lai, “Schedule execution for two-machine flow-shop with interval processing times,” *Mathematical and Computers Modelling*, vol. 49, 2009, pp. 991-1011.

[22] Z. Bouyahia, M. Bellalouna, P. Jaillet, and K. Ghedira, “A priori parallel machines scheduling,” *Computers & Industrial Engineering*, 2009, doi: 10.1016/j.cie.2009.11.009

[23] I. Mahdavi, B. Shirazi, and M. Solimanpur, “Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems,” *Simulation Modelling Practice and Theory*, vol. 18, no. 6, 2010, pp. 768-786, doi: 10.1016/j.simpat.2010.01.015.

[24] A. J. Davenport, J. C. and Beck, “A survey of techniques for scheduling with uncertainty,” unpublished. Available online from <http://www.eil.utoronto.ca/profiles/chris/gz/uncertainty-survey.ps> or <http://eil.utoronto.ca/profiles/chris/chris.papers.html>

[25] W. Herroelen, and R. Leus, “Project Scheduling under uncertainty: Survey and research potentials,” *European Journal of Operational Research*, vol. 165, 2005, pp. 289-306.

[26] H. Aytug, M. A. Lawley, K. McKay, S. Moha, and R. Uzsoy, “Executing production schedules in the face of uncertainties: A review and some future directions,” *European Journal of Operational Research*, vol. 161, 2005, pp. 86-110.

[27] J. Mula, R. Poler, J. P. García-Sabater, and F. C. Lario, “Models for production planning under uncertainty: A review,” *International Journal of Production Economics*, vol. 103, 2006, pp. 271-285.

[28] N. Al-Hinai, and T. Y. ElMekkawy, “Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm,” *International Journal of Production Economics*, vol. 132, 2011, pp. 279-291, doi: 10.1016/j.ijpe.2011.04.020.

[29] I. Kacem, S. Hammadi, and P. Brone, “Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 32, no. 1, 2002, pp. 1-13.

[30] K. Sörensen, “Tabu searching for robust solutions,” in *Proc.4th metaheuristics international conf.*, Porto, Portugal, 2001, pp. 707-712

[31] D. Y. Lee, and F. DiCesare, “Scheduling flexible manufacturing systems using petri nets and heuristic search,” *IEEE Transactions on Robotics and Automation*, vol. 10, no. 2, 1994, pp. 123 – 132.

[32] P. Brandimarte, “Routing and scheduling in a flexible job shop by tabu search,” *Annals of Operations Research*, vol. 41, 1993, pp. 157–183.

[33] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet Physics – Doklady*, vol. 10, 1966, pp. 707-710.

Nasr Al-Hinai is an assistant professor at the department of Mechanical and Industrial Engineering, Sultan Qaboos University (SQU), Oman. He received his B.Eng in Mechanical Engineering from SQU in 2001, his M.Sc. from the department of Mechanical, Manufacturing, and Aerospace Engineering, UMIST, UK in 2003 and his PhD from the Department of Mechanical and Manufacturing Engineering, University of Manitoba, Canada, in 2011. He worked as a demonstrator in the department of Mechanical and Industrial Engineering, SQU, in 2001, as a lecturer in 2003, and then rejoined the department as an assistant professor in 2011. His research interest is in the area of scheduling optimization, and operations research.

Tarek Y. ElMekkawy is an associate professor at the department of Mechanical and Industrial Engineering, University of Manitoba. He received his B.Sc. and an M.Sc. from the Department of Mechanical Design and Production, Cairo University, Egypt in 1990 and 1994, respectively. He received his PhD from the Department of Industrial and Manufacturing Engineering, University of Windsor, Canada, in 2001. He has joined the University of Manitoba (UM) in 2003 after working for two years in automotive industry, Ontario, Canada. His research interest is in the areas of scheduling optimization, and applications of operational research and industrial engineering techniques to improve healthcare delivery. He has published many papers in international journals, such as *IJPR, IJPE, IJCIM, IJAMT, IJOR,* and *CIRP Annals*.