

AN EMPIRICAL ANALYSIS OF THE RELATIONSHIP BETWEEN THE INITIALIZATION METHOD PERFORMANCE AND THE CONVERGENCE SPEED OF A META-HEURISTIC FOR FUZZY JOB-SHOP SCHEDULING PROBLEMS

¹IMAN MOUSA SHAHEED, ²SYAIMAK ABDUL SHUKOR, ³ERNA BUDHIARTI NABABAN

^{1,2} Industrial Computing Research Group, Centre for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor, Malaysia

³ Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia.

E-mail: ¹eman_musa21@yahoo.com, ²syaimak@ukm.edu.my, ³ernabr@usu.ac.id

ABSTRACT

Numerous studies mentioned that the quality initial population can affect meta-heuristics convergence speed, these studies are often theoretical. However, the functionality of the initial population is extensively ignored. This overlooking may due to the literature lack of statistical evidence on the relationship between the initialization method performance and a meta-heuristic convergence speed. Therefore, this study statistically investigated aforementioned relationship by conducting an experiment and used population quality and best error rate (BRE) to gauge the performance of the state of the art initialization methods for Fuzzy Job-Shop Scheduling Problems (Fuzzy JSSPs), namely, random-based and priority rules-based methods. Thereafter, this initialization approach utilised to initiate a memetic algorithm (MA). CPU time was used to compute the MA time to reach the lower bound of 50 different sized Fuzzy JSSP instances. A Spearman's test was operated to measure the intended correlation. As a result, there was effective negative association between the initial population quality and the MA convergence speed. While, there was a dominant positive relationship between the BRE and MA convergence speed. Consequently, it is highly recommended to develop advanced initialization approaches that can generate high-quality initial population, which consists of most favourable or near to best possible solutions.

Keywords: *Fuzzy Job-Shop Scheduling Problems, Initialization Methods, Memetic Algorithm, Population Quality, Best Relative Error, Convergence Speed.*

1. INTRODUCTION

The scheduling operations is one of the major crucial problems in production and manufacturing organizations along with in the information processing organizations [1]. In the real manufacturing systems, provided machinery and collection of employments with a variety of predestined directions through the machinery equipment, the purpose is to bring together a schedule of duties that will reduce the definite decisive factors such as makespan, limit delay, and measuring punctuality. In this approach various information, such as things associated with due dates, operating times and technical restriction, are not essentially deterministic. These inconveniences could be mainly due to the key factor which

comprises a modification in the purchaser order, an adjustment in the accessibility date, unable to get raw materials delivery and breakdown of engine. This is put behind to belief that employees are accountable for these inconveniences rather fuzzy setting up of information or Fuzzy JSSPs [2]. The Fuzzy JSSP has been recognized as the resolution for this type of NP-hard issues which requires successful manufacturing scheduling system to improve the efficiency and effective machine deployment [3].

In recent years, the adoption of population dependent meta-heuristic technique such as particle swarm optimization (PSO) by Li, Pan [4], genetic algorithm (GA) by Palacios, González [5], and memetic algorithm (MA) by Palacios, Puente [6] has given better outcomes for Fuzzy JSSPs than the conventional approaches or heuristic algorithm.



The Meta-heuristic approaches creates more logical schedules in a realistic computational period, and could obtain near to best possible resolution effortlessly [7]. However, Palacios, Puente [6] found that MA has advanced other algorithms for Fuzzy JSSPs in reaching the optimal with a high convergence speed. The convergence speed measures how long time an algorithm takes until solutions have reached a certain similarity and the population is converged [8].

The life cycle of MA comprises of the numerous segments like population initialisation, which occurs only once. Followed by the genetic approach which coordinates with a number of generations (i.e. assortment, intersect, alteration and termination constriction). In each phase a fresh generation is created from the preceding one by imposing the typical genetic operative. Tabu search is employed to all schedules created by either initialization approach or by the GA. The consequent gene is restored from the enhanced schedule acquired by TS so its features can be transmitted to the succeeding brood.

Population initialization is a critical job in population dependent meta-heuristic problem-solving operations [9]. According to [10, 11], the performance of population defined variables is very significant than every phases of function in population-based meta-heuristics to mainly amplify the effectiveness and to obtain the adjacent value for the necessary and most favourable resolution.

In this context, numerous studies mentioned that the goodness of the initial population can affect meta-heuristics convergence speed. According to Maaranen, Miettinen [12], population in each iterative improvement process by MA relayed on its previous population and, lastly, on the opening population, thus, the original population has a particular role on increasing the computation and to attain the concluding best possible resolution. This role has been advocated by the superiority of the original population, in particular, can amplify the union speed and develops the value of the ultimate resolution. However, these studies are often theoretical.

On the other hand, researchers such as [13-15] suggests that this role of the initial population is widely overlooking. Recently, Abdullah and Abdolrazzagh-Nezhad [7] stated that initial population is given little attention by the current researches.

To our knowledge, the relationship between the initialization method performance and a meta-

heuristic convergence speed has not been statistically investigated yet, and this literature lack of such statistical evidence could be one reason of this ignoring.

Therefore, the core purpose of this study was to statistically explore the correlation between the initialization method performance and a meta-heuristic convergence speed. To do so, an experiment was conducted to evaluate the performance of the state of the art population initialization practices used for Fuzzy JSSPs based on a feature of performance factors, i.e. the BRE and population quality. Thereafter, the relationship of the performance of these initialization practices with the convergence speed of the improvement algorithm used to solve Fuzzy JSSPs, in this study, MA by Puente, Vela [16] was examined.

This study provides the first explicit investigation for the effectiveness of the quality of initial population to the performance of meta-heuristic algorithms. In another words, the findings of this study could serve as the unique statistical evidence to the intended relationship and therefore encourage the researchers to pay more attention to the development of advanced initialisation approaches including heuristic-based methods.

Here are the different sections the study has been categorized into: Section 2 reviews relevant literature. Section 3 present the description of the experimental design used in this study. The presentation of results and its discussion are in Section 4. The conclusion presented in Section 5.

2. BACKGROUND

2.1 Population-based Meta-heuristic

Glover [17] was the first to come up with the concept of Meta-Heuristics algorithms. According to the definition published in Blum and Roli [18], Meta-Heuristics are intelligent routes for solution space exploration and optimal solution exploitation. Meta-Heuristic algorithms have played a significant role in solving Fuzzy JSSPs. Most recent literature review by Behnamian [19] shows that Meta-Heuristic algorithms receives major attention than other techniques in the fuzzy scheduling environment with more than 70% of the existing studies.

Numerous population-based meta-heuristic algorithms have been developed for Fuzzy JSSPs [4, 20-25]. In these algorithms, multiple solutions were maintained that interact with each other to explore the search space. The implementation process of one Meta-Heuristic for JSSPs consists of three steps – pre-processing, initialisation and improvement (See Figure 1).

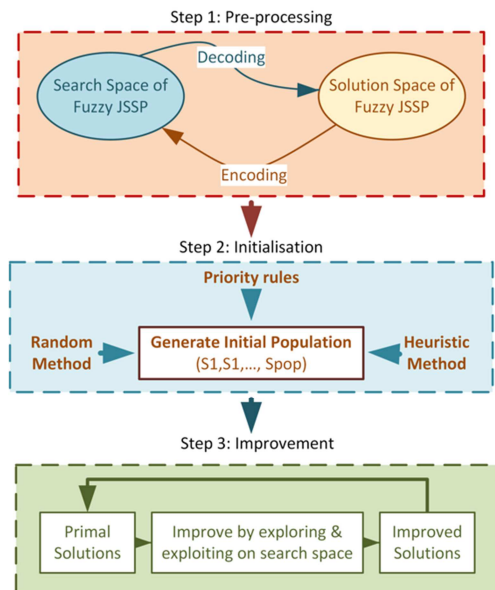


Figure 1. Implementation of one Meta-Heuristic for Fuzzy JSSPs [7]

Pre-processing is then broken-down into two sub-tasks – encoding the solution space and preparing a decoder algorithm for generation of a viable fuzzy schedule in conformity with the encoded point. During the initialisation process, a primary population needs to be produced. Intelligent algorithms based on exploration of the Fuzzy JSSP solution space are then implemented

with the sole purpose of improving the original solutions.

Population initialisation plays a key role in the quality of the final resolution and computation speed [26-28]. Very few researchers have taken up this matter [13, 29] and the advancement of population initialisation is not particularly popular either [7]. The following sections concentrate on exactly that – the initialisation step and the most commonly used initialisation methods in Fuzzy JSSP resolution.

2.2 Job-shop scheduling Problems with fuzzy processing time

In actual world production organization there are varieties of worries associated with the (a) consumer demands; (b) discharge dates, (c) operational and shipping symbolize a spanner in the mechanism of manufacturing process. These problems can be mainly due to factors which involve a transformation in the purchaser bid, an adjustment in the accessibility date, malfunction in the delivery system of raw materials, and breakdown of engine.

The Fuzzy JSSP can be describe as follow: There are a collection of n jobs to be put into practice on m machinery. Every task i comprises of a series of n_i functions $O_{i,j}, j = 1, 2, \dots, n_i$. Each one of routing should to be resolute to the entire operations of a job. The implementation of $O_{i,j}$ needs one appliance out of a collection of given apparatus $M_i \subseteq M$. The dispensation period of the $O_{i,j}$ on device k is stands for the Two Fuzzy Numbers (TFN) $t_{i,j,k} = (t_{i,j,k}^1, t_{i,j,k}^2, t_{i,j,k}^3)$, likewise, the fuzzy finishing period of task $O_{i,j}$ is correspond to TFN $C_{i,j} = (C_{i,j}^1, C_{i,j}^2, C_{i,j}^3)$, In this case $C_{i,j}^1$ is the least achievement instance, $C_{i,j}^2$ is the most possible finishing time and $C_{i,j}^3$ is the prime finishing time. The main objective of Fuzzy JSSP is to find out both the task of machinery for the every functions and the progression of the functions on every machines to decrease a definite purpose, such as the subsequent highest fuzzy finishing time:

$$C_{max} = \max_{i=1,2,\dots,n} C_i, \quad (1)$$

where C_i is the fuzzy finishing time of task i .

In this context, some fuzzy variable functions are requisite to create a schedule. These functions comprise the additional function, the max function of the two fuzzy variables, and the positioning process of fuzzy variables. Addition process is

utilized to estimate the fuzzy finishing time of an operating process. Max function is utilized to find out the fuzzy initial time of an functional process. And the positioning process is used to evaluate the utmost fuzzy finishing time. For the two triangular fuzzy variables (TFNs), $A = (a^1, a^2, a^3)$ and $B = (b^1, b^2, b^3)$, their adding up is shown by the following principle:

$$A + b = (a^1+b^1, a^2+b^2, a^3+b^3) \quad (2)$$

In this report, the subsequent principles are implemented to grade fuzzy numbers to make a possible schedule.

Step1: The greatest number $C_1(A) = (a^1+2a^2+a^3)/4$ will be chosen as the first criterion to rank two TFNs.

Step2: If two TFNs have the same $C_1(A)$, $C_2(A) = a^2$ is used as a second criterion.

Step3: If C_1 and C_2 are identical, $C_3 = a^3 - a^1$ will be chosen as a third criterion.

The above principle executes better on the fuzzy scheduling. Thus, the maximum of two TFNs A and B is approximated by this principle in this report. That is, if $A > B$, then $A \vee B = A$ else $A \vee B = B$.

2.3 Population Initialization Methods for Fuzzy JSSPs

For Meta-Heuristic algorithms to solve Fuzzy JSSPs, the earlier works are framed into three processes for primary JSSPs solution construction steps. These are Pre-processing, Population initialization, and Improvement, and heuristics [7].

Population initialization techniques have various types of different characteristics. Currently, we recognized two types of initialization methods in the Fuzzy JSSPs literature, which are Random-based and priority rules-based methods. This section provides a brief evaluation on these population initialization approaches for the Fuzzy JSSPs.

2.3.1 Random-based Methods

With the absence of prior information on the population, The construction of random population is considered uncomplicated [7]. The available rationale exposes that random approaches which are ideal choice of the majority of researchers for creating the initial population. This my due to these methods effortless implementation, particularly concise computation period and effective diversification of the primarily created phases in the

challenges of search space. Below is review to the selected papers that used random-based methods for the Fuzzy JSSPs initialisation.

In the research of Fortemps [30] he started from an initial solution obtained in the following way: they consider each task successively in an arbitrary order and put their operations on the list of operations to the appropriate machine, always after the previous ones. Then, to progress to other feasible sequencings, they revert the orientation of some critical arcs of the disjunctive graph. The translation into an element of the solution space is straightforward. However, this gives us a first solution, which is presumably very poor.

In the work of Lin [31], Johnson's constructive algorithm has been used to generate solutions. This algorithms, however, randomly orders the jobs. Tavakkoli-Moghaddam, Safaei [32] initiate the proposed fuzzy-neural technique for constriction contentment of a generalized job shop scheduling problem (GJSSP) fuzzy release times by the following way: (a) obtain the release period of process as a triangular fuzzy number. (b) Assign every function at random to a functioning machine. (c) Start over $W_{ST}, W_{PT}, \epsilon, \delta$ and $t=0$. (d) Set $S_{ikp}(0) = U(0, S_{Max}) \forall i, k, p, O_{ikp} \in R_p$. In this case $U(a, b)$ is a continuous uniform distribution and S_{Max} denotes a large arbitrary number. (e) Set $T_{ikp}(0) = M_{ikp} \forall i, k, p, O_{ikp} \in R_p$.

The job shop scheduling problem with uncertain time period represented as triangular fuzzy numbers and used random numbers to generate initial solutions for their experiment. However, this procedure may produce illegal solutions. In order to eradicate the manufacturing of illegitimate gene, in the studies of [24, 33-35] the N chromosomes produced were unpredictable as the opening population, and every chromosome is involves of $n \times m$ usual numeral codes in $[1, n]$. N is the population integer. The turn unsystematic number is utilized to construct natural number in $[1, n]$ and tracing of times of the number, which is generated.

Moreover, to generate feasible fuzzy schedules, works [36-43] developed the G&T algorithm, which were then entrenched with unsystematic assortments to creates initial fuzzy schedules. Succeeding to this approach, the fuzzy schedules were entrenched with unsystematic assortments to acquiesce preliminary fuzzy schedules. This is to come up with viable fuzzy schedules. Song, Zhu

[44] used the extension of G&T algorithm by [45] to form the initial swarm.

[5, 46] proposed a random initialisation method based on G&T algorithm. To produce opening resolution functions were scheduled in an incorporation method except intriguing benefit of the elasticity, by defining a feasible insertion interval for an operation o_{ij} in a machine $M_k \in M(o_{i,j})$ to be a time interval $[t^S, t^E]$ in which machine M_k is inactive and such that o_{ij} can be develop within that period of time devoid of contravene priority limitations. Gao, Suganthan [47] projected a new heuristic, called as MinEnd heuristic, for initialising population. In this approach of heuristic, the function series is determined erratically. The dispensing appliance for every function is allocated based upon the functional series. The process of this approach is revealed as follows: Stage 1: Every function of each task is mixed up erratically to gain an operational sequence. Stage 2: Fixing of OS. Make confirm that the function of the similar work can assure the processing of the main concern Stage 3: For every function $O_{i,j}$ in OS, estimate the fuzzy finishing time on all selectable machines. Stage 4: The device with least amount fuzzy finishing time is chosen for processing function $O_{i,j}$.

2.3.2 Priority rules-based methods

The Priority rule is an uncomplicated heuristic employed mainly for sequencing jobs in a scheduling issue. At this instant when a decision for sequencing required to be prepared, the priority rules will amplify the process in a row of the device. Subsequently, the task with the maximum concern is the one at the next to be functional at the consequent device [48]. According to Abdullah and Abdolrazzagh-Nezhad [7], priority rules came second after random techniques in its usage for Fuzzy JSSPs initial population generation. This popularity can be outlined as they are effortlessly implemented and is least concern about calculations and are never bogged behind by period obscurity.

In Itoh and Ishii [49], they used an extended due-date rule that Ordering the objective jobs according to their due date. Kuroda and Wang [2] used a kind of slack priority rule considering the fuzziness of the due date and processing times. Petrovic and Fayad [50] used the Early Due Date (EDD) priority rule's sequencing capabilities to generate a population. As the name suggests, the EDD rule processes jobs, based on their respective due dates, i.e. the ones with the earliest due dates are handled with top priority. Fayad and Petrovic

[33] altered the process and one of four rules was selected on random basis for sub-chromosome handling: EDD, SPT, LPT, and LRT. Lei [51] used a G&T method, which in conjunction with five priority rules cancels the i^{th} conflict occurring in machine M_j . This is to negate the potential incorrect representation, caused by the usage of a random range.

3. LITERATURE REVIEW

Fuzzy Job Shop Scheduling Problems (Fuzzy JSSPs) considered as one of the most popular research topics in this domain due to its potential to dramatically decrease the costs and increase the throughput.

Since Fuzzy JSSP is an NP-hard problem, there has been a growth of interest in the development of meta-heuristic algorithms to solve it, such as; the adoption of population-based meta-heuristic algorithms like particle swarm optimization (PSO), genetic algorithm (GA), and memetic algorithm (MA) has led to better results for Fuzzy JSSPs than the traditional dispatching or heuristic algorithm [7].

Most recent literature review by Behnamian [19] shows that Meta-Heuristic algorithms receives major attention than other techniques in the fuzzy scheduling environment with more than 70% of the existing studies. However, a major disadvantage is the lengthy calculation period, particularly in the case of such a challenging solution space [9, 26].

Population initialisation considered as a crucial task in population-based meta-heuristic algorithms [9]. According to [10, 11], the performance of population initialisation is more important than any other stage of operation in population-based meta-heuristics to largely increase the efficiency and to get the nearest value to the required optimal solution.

In this context, numerous studies mentioned that the goodness of the initial population can affect meta-heuristics convergence speed. According to Maaranen, Miettinen [12], population in each iterative improvement process depend on its previous population and, lastly, on the initial population, thus, the initial population has a special role on speed up the computation and obtain the final optimal solution. This role has also been advocated by [27, 28, 52] who mentioned that the quality of the initial population, in particular, can accelerates the convergence speed and improves the quality of the final solution.

However, relevant literature reviewed in this section highlight the idea of that the final solution quality and convergence speed can be influenced by the goodness of the initial population in terms of its quality and diversity.

Surprisingly, this relationship has never been investigated quantitatively. This could be one reason why this role of the initial population is widely overlooking and given little attention by the current researches [7, 13-15]. Apart from the theoretical evidences provided by the literature. A quantitative study to establish this correlation could be an essential factor to encourage the researchers pay attention to the importance of the initialisation process.

4. EXPERIMENTATION DESIGN

This segment mainly represents the experimental design required for identifying the performance features of different population

initialisation method for MAs to solve Fuzzy JSSPs. The segment is focused on the experiment environment, performance evaluation method and tools for analysing the statistics.

The experimental framework is shown in Figure 2 in which the preliminary population created with altered population initialization technique are prepared to associated with the precise intersect operator unless the termination condition is fulfilled.

The individuals' population of resultants, which is gained from the analogous test setting, is calculated with the distinct performance factor. The MA limitation and the resultant ethics are shown in Table 1.

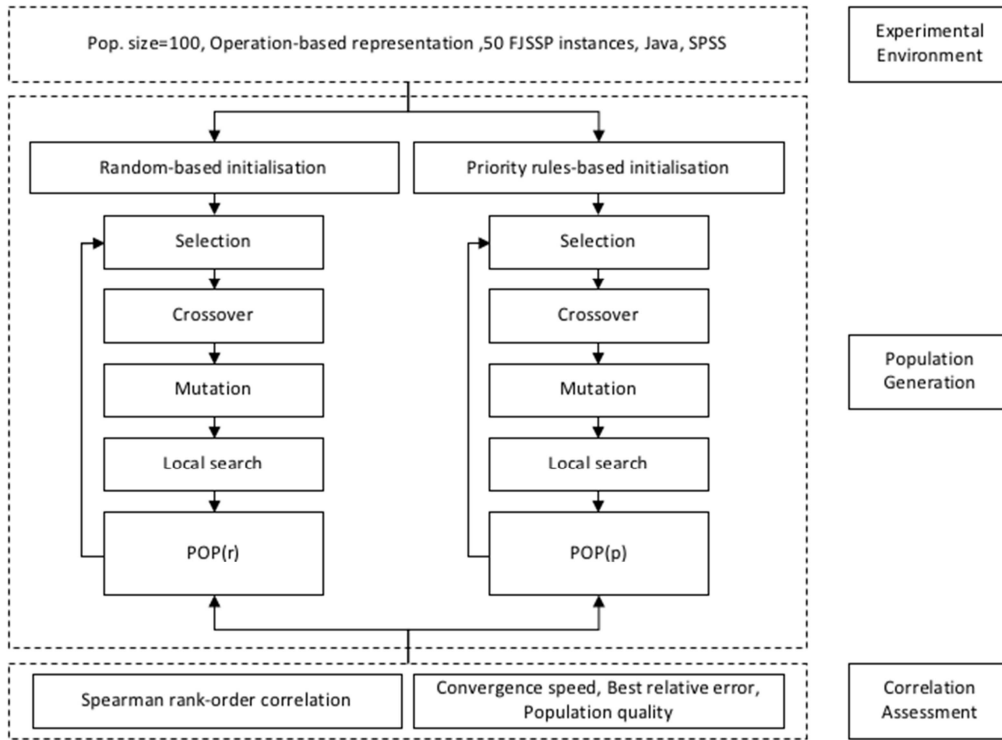


Figure 2: Experimental Design

Selection approach is carried on to make sure that the healthy population in every age group are passed away to the other age group for avoiding the substitution of the healthiest population with the weakest individual in the next age group.

Table 1: MA Configuration Parameter For Experiment

No.	Parameter	Value/technique
1.	Population size	100
2.	Generation limit	500,000
3.	Initialization technique	Random-based, priority-rules-based



4.	Crossover method	Precedence Order-based Crossover (POX)	preserving Crossover	environmental situation for calculating the performance with accuracy. The primary population volume for chromosomes will be 100. This volume has been just applied by [5]. In this experiment, the independent variable used is the initialization approach. This definite variable are of two values, which are the state-of-the-art initialisation methods for Fuzzy JSSP, namely, Random-based method by Palacios, González [5] and Priority rules-based method by Lei [51]. At the other situation, the continuous dependent variables are the individual class and variety, the error rates (i.e., finest comparative error, standard comparative error, and Worst comparative error). In this, the continuous variable has a direct relationship with the independent variable. The measurement method for each dependent variable is discussed in section 3.2. All algorithms will be applied and run in JAVA program language and also in the Eclipse of the Luna with 4.4 IDE [56], on the processor intel core with 2.5 GHz and RAM with 4 GB.
5.	Crossover probability	0.6		
6.	Mutation method	TS-based local search		
7.	Mutation probability	1		
8.	Selection	2 individuals		
9.	Termination condition	Generation limit		

For every method, the implementations are carried out for 10 times and the standard of all the cases has been accepted for the experimental analysis. In the testing, the operator crossover used is the extension of the well-known Precedence preserving Order-based Crossover (POX) suggested by Kyung-Mi, Yamakawa [53]. The POX which has been initiated is uncomplicated and also has useful features chosen by operator crossover which is valuable in combining two or more population for transferring the superior information to the next generation. POX produces new chromosomes from the two chromosomes which already exist in the following way: 1) An operation O_i is selected randomly from one parent P_1 , 2) Locate all O_i preceding operations PD_i , 3) Place the operations of PD_i , from P_1 into child C_1 , at the same positions, 4) The operations which is left is placed in child C_1 , in the same way which emerge in the other parent P_2 .

While the local search algorithm (TS) will substitute the mutation operation, as recently used by Caumont, Lacomme [54]. Moreover, an implicit mutation by Tabu search as has recently been used by Caumont, Lacomme [54] is applied. It is the cause for not choosing any explicit mutation operator. As a result, the factors used in the experimental study are significantly simple, as the crossover probability is placed to 1 and mutation probability doesn't require specification. According to the situation, results which are parallel to those obtained with the minor crossover probability and a lower probability applied by the mutation operator are achieved. There are some instigator, for example González, Vela [55], have previously observed that a mutation operator has no significant role in genetic algorithm hybridized with local search.

4.1 Experimentation Environment

Experiment was performed on various individual initialization techniques over same

4.1.1 Fuzzy Objective Function

The scheduling objective, in this study, is to locate a possible program to reduce the expected fuzzy makespan. Hence an estimated function is necessary to analyse the makespan of programme fixed in chromosomes. The makespan is the most significant amongst the sum of the initial time of the final operation in every machine and its operating time. The makespan can be established through removing the information regarding the procedure processing sequence for every machine and the processing opening time of operation from a possible chromosome.

4.1.2 Individual Representation

In order to codify chromosome we chosen a direct, representation mainly based on operation to be used [57] to represent a solution in the search space. This scheduling encoding schema relies on the operations sequence, however, the work are particularly used as the similar decimal code, and after that the read based on their appearance order within certain chromosome. In all the illustration types, operation-based representation is in the first position [7]. This representation's decoder works in two main stages. The first one involves the transformation of the enciphered point to a record of well-coordinated operations. The second one applies a one-pass heuristic in order to generate a schedule. In this study, an upgraded G&T algorithm

version by Hu, Yin [58] is used in order to apply fuzzy operators for a fuzzy schedule generation, thus making the algorithm the actual fuzzy decoder.

4.2 Population Generation and Improvement

In any meta-heuristic for Fuzzy JSSPs, the initial stage is to generate initial solutions. After that, the improvement algorithm performs over the numbers of the generations. In every respective phase, a fresh generation is put together from the preceding one by implementing the standard genetic operative. This generation process continues until the termination condition is satisfied.

This study adopted the memetic algorithm (MA) [59] as the improvement algorithm. There are several MAs, which have been considered viable for Fuzzy JSSP application [5, 6, 16, 60-63]. However, most existing MAs to solve Fuzzy JSSPs have randomly initiated. Thus, this study will show the undiscovered implementation of MA with priority rules as initialization method.

MAs have a proven record for dealing with optimisation problems related to scheduling [64]. This is the reason behind the select of MA in this research especially that merge with the genetic algorithms (GA) and Tabu Search (TS). The flowchart of the memetic algorithm can be observed in Figure 1.

In this study, a machine-block, a sequence of operations of the same job processed successively, is used to obtain the neighbourhoods. Theoretical thoughts demonstrate that swapping neighbourhood critical operations in the internal part of machine-block cannot lead to refine schedules makespan Van Laarhoven, Aarts [65]. Instead, it is recommended that swapping one operation at the end of one block with another one at the beginning of the next block can help to obtain good neighbourhoods [65]. Defining the critical paths can identify significant operations.

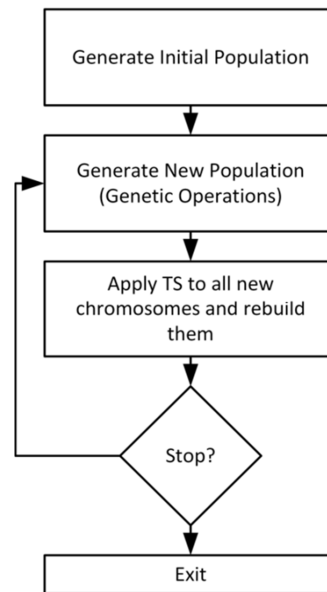


Figure 1: Memetic Algorithm Flowchart

A significant path is a progression of significant functionalities that begin from time zero to the completion time. Following the schedule from its completion time to the time zero can discover a critical path. Identifying the critical machines, the machines that are working until the last time of the schedule, can do this. The completion time of the critical machines consider as the makespan of the schedule. Therefore, every impediment in processing the last operation of the vital machines will defer the whole schedule's makespan. Thus the critical machines last operations consider as critical operations. To identify other critical operations, the following cases are applicable: 1) the machine-precedent operation: if the machine-precedent operation completion time and the critical operation start time are equal (No idle time is found), 2) The job-precedent operation: if the critical operation start time and the job preceding operation finish time are equal, 3) Both case 1 and case 2. This procedure continues until time zero is reached.

4.3 Performance Factors

In this research, the population quality, the best relative error (BRE), and the improvement time are used as the analysis criteria. This is to assess the initialization method performance. These terms, however, is discussed in the next sub sections.

4.3.1 Best Error Rate

The performance of the starting procedures is measured in the Fuzzy JSSPs through an ideal purpose with lowering the possibility of the fuzzy makespan as to be the smallest achievable level, at

the condition of the ideal solution created by both the starting method have been planned and comparing the several learning outcomes [14, 66, 67]. As a result, the features shown are measured in this study. The best comparative error (BRE) is measured as in Li and Yin [68]:

$$BRE = \frac{(Sol_{best} - Best_{known})}{(Best_{known})} \times 100 \quad (3)$$

$Best_{known}$ is the appropriate explanation in the narration. n is the size of the population. The Sol_{best} is the ideal explanation generated through initialization method. BRE is applied to reply any question of the closeness of the best solution of the formed population as per the best known solution exist. The least value of BRE demonstrates that of the appropriate solution is near with the finest solution that exist, intended for the particular problem instanced.

4.3.2 Population Quality

For calculating the quality of initial population, the standard of the meeting rate of solutions in the initial population produced will be used [69]. Victor Paul, Moganarangan [70] did use it to measure initial population quality. It is described as the standard of the junction rate of solution with the primary population created [69]. It is demonstrated as below:

$$\text{convergence of average(\%)} = \left(1 - \frac{\text{fitness of average} - Best_{known}}{Best_{known}}\right) \times 100 \quad (4)$$

Where the fitness average is the fitness average value of the solution in the population; $Best_{known}$ are the best known solution of the parallel instances. The above feature is implemented to calculate the value of the population created as the standard of the population as given in Eq. (4). The primary population having good standard junction also increases the performance in terms of the convergence of time and can also help to investigate the explore space in better way [5].

4.3.3 Convergence Time

In order to compute the convergence time for MA, the time of the CPU is measured using the JAVA through the function using the *System.nanoTime()*, that proceeds with the running time in the Nano seconds and consequently converts into second. The purpose is use exactly prior to generating the first age group and entitled yet again after completion of each succeeding age group [71].

4.4 Data Sets

We considered 50 Fuzzy JSSP benchmarks instances, that are mainly used for testing the effectiveness of algorithms. The selected benchmark instances are applied as *test-beds* to measure the activities and measure the efficacy of the improved heuristic approach contrast with the existing initialization methods for Fuzzy JSSPs. Table 2 shows the test-bed trouble, their size and the highest maximum standards for the presentation analysis.

Table 2: Dataset Description

No	Dimension ($M \times N$)	Instances	# of instances
1.	6×6	S6.2	1
2.	5×10	LA01	1
3.	5×15	LA08	1
4.	10×10	ABZ5, LA19, ORB02, S10.2, S10.4, ABZ7, ABZ8, ABZ9	8
5.	10×15	LA21, LA23, LA24, LA25, Lei02	5
6.	10×20	LA29, LA30	2
7.	15×15	LA36, LA37, LA38, LA39, LA40, TA01, TA02, TA03, TA04, TA05, TA06, TA07, TA08, TA10	14
8.	15×20	TA13, TA14, TA15, TA16, TA17, TA19	6
9.	20×20	TA23, TA24, TA25, TA26, TA28, TA29, YN01, YN02, YN03	9
10.	15×30	TA34	1
11.	20×50	TA67, TA68	2
TOTAL			50

The particular issues are of various size and rigidity, range from 6×6 (6 jobs and 6 machines) to 20×50 (20 machines and 50 jobs) hence the performance of the methods can be applied on different datasets. Thanks to the effort of Palacios, Puente [6], who explicitly provide Fuzzy JSSP researchers community with all datasets with their lower bound information.

4.5 Spearman's Correlation Analysis

There are many statistical tests that can be used in order to analyse the relationship between two variables. In this study, we adopted the Spearman rank of order of correlation (often referred as the Spearman's correlation) [72]. It measures the coefficient, r_s or ρ (pronounced "rho"), that is used to evaluate of the strength and way of the relationship between two variables. According to Cohen [73] guidelines, the value of the coefficient correlation (r) which ranges from -1.00 to the 1.00. It will help in indicating the power of the



association between any two variables. An association of 0 indicate negative association at all, a correlation of 1.0 refers an ideal positive association, and a value of -1.0 specify an ideal negative correlation. the relationship is considered “poor” when ($r = 0.1$ to the 0.29), “medium” when ($r = 0.3$ to the 0.49), and “strong” when ($r = 0.5$ to the 1.0). In this study, the association between BRE and convergence time in this study, as well as, the association between population quality and convergence time in this study will be measured.

5. EXPERIMENT AND RESULTS ANALYSIS

In this section, the correlation of the MA with Random-based and priority-based population initialisation techniques in solving Fuzzy JSSP

occurs under parallel experimental setup which is discussed according to the suitable set of performance factors as discussed in Section 3.2. The particular correlation analyses are as follows:

5.1 The Correlation Between the Best Error Rate and Convergence Speed

A Spearman's rank-order of correlation was used to measure the correlation between BRE as a performance factor and the convergence time of MA. The results shown that there was a strong positive correlation among BRE with the convergence time of MA for different types of initialization method: the Random-based method, $r_s(498) = 0.614$, $p < .0005$, and Priority rules-based method, $r_s(498) = 0.656$, $p < .0005$. See Figure 3.

Correlations

Initialization Method			Best Relative Error	Convergence Time
Spearman's rho	Random	Best Relative Error	Correlation Coefficient Sig. (2-tailed) N	1.000 .614** 500
		Convergence Time	Correlation Coefficient Sig. (2-tailed) N	.614** 1.000 500
	Priority rules	Best Relative Error	Correlation Coefficient Sig. (2-tailed) N	1.000 .656** 500
		Convergence Time	Correlation Coefficient Sig. (2-tailed) N	.656** 1.000 500

** . Correlation is significant at the 0.01 level (2-tailed).

Figure 3: SPSS Result of Spearman's Rank-Order Correlation Between the BRE and Convergence Time over Different Initialization Methods

5.2 The Correlation Between the Population Quality and Convergence Speed

A Spearman's rank of order of correlation was used to measure the association between population quality as a performance factor and the convergence time of MA. The results shown strong

negative correlation is present between population quality with the convergence time of MA for different types of initialization method: the Random-based method, $r_s(498) = - 0.522$, $p < .0005$, and Priority rules-based method, $r_s(498) = - 0.514$, $p < .0005$ (See Figure 4).

Correlations

Initialization Method				Population Quality	Convergence Time
Spearman's rho	Random	Population	Correlation Coefficient	1.000	-.522**
		Quality	Sig. (2-tailed)	.	.000
			N	500	500
	Convergence Time	Population	Correlation Coefficient	-.522**	1.000
		Quality	Sig. (2-tailed)	.000	.
			N	500	500
Priority rules	Random	Population	Correlation Coefficient	1.000	-.514**
		Quality	Sig. (2-tailed)	.	.000
			N	500	500
	Convergence Time	Population	Correlation Coefficient	-.514**	1.000
		Quality	Sig. (2-tailed)	.000	.
			N	500	500

** . Correlation is significant at the 0.01 level (2-tailed).

Figure 4: SPSS Result Of Spearman's Rank-Order Correlation Between The Population Quality And Convergence Time Over Different Initialization Methods

6. CONCLUSIONS

In this paper, the state-of-the-art population initialization techniques for Fuzzy JSSPs, namely, random-based and priority-based system have been learned and examined. The practical capacity of those initialization technique is assessed by means of the performance criteria such as best relative error rate and population quality.

In addition to the computational time computed for MA started by those initialization methods. The experiments were performed on various size benchmark Fuzzy JSSP instances obtained from literature.

Spearman's correlation coefficient was implemented to evaluate the force and the path of the relationships between best relative error rate and MA convergence speed as well as between the initial population quality and MA convergence speed. From the experimental analyses it can be confirm the hypothesis in that there is a strong relationship between the initialization method performance and a meta-heuristic convergence speed.

The established relationship may influence the researchers' tendency to move on to develop more advanced initialization methods that may overcome the existing shortcomings and generate high-quality initial population, which consists of optimal or near optimal solutions. This is to reduce the computational time of the improvement algorithm.

This investigation of the relationship has limited only on a two-value independent variable, which includes the random and priority rules initialization methods. However, these are the existing method in the field of fuzzy JSSPs. Based on our superior deem, including more types of population initialization method such as heuristic-based and chaotic-based methods can strength the intended relationship. Therefore, considering more than two values is suggested in future works.

Moreover, this study used the initial population quality as a performance criterion of the initialisation method. However, the diversity of the initial population considered as an important factor in the meta-heuristics domain. To our knowledge, initial population diversity has never been explicitly measured for Fuzzy JSSPs initial population. Therefore, this important performance criterion should be taken into account in the future. The relationship between population diversity and meta-heuristics performance need to be investigated too.

REFERENCES:

[1] Pinedo, M.L., *Scheduling: theory, algorithms, and systems*. 2012: Springer Science & Business Media.
 [2] Kuroda, M. and Z. Wang, *Fuzzy job shop scheduling*. International Journal of Production Economics, 1996. 44(1-2): p. 45-51.



- [3] Roshanaei, V., *Mathematical Modelling and Optimization of Flexible Job Shops Scheduling Problem*, in *Industrial and Manufacturing Systems Engineering*. 2012, University of Windsor.
- [4] Li, J., et al., *Solving fuzzy job-shop scheduling problem by a hybrid PSO algorithm*, in *Swarm and Evolutionary Computation*. 2012, Springer. p. 275-282.
- [5] Palacios, J.J., et al., *Genetic tabu search for the fuzzy flexible job shop problem*. *Computers & Operations Research*, 2015. **54**(0): p. 74-89.
- [6] Palacios, J.J., et al., *Benchmarks for fuzzy job shop problems*. *Information Sciences*, 2016. **329**: p. 736-752.
- [7] Abdullah, S. and M. Abdolrazzagh-Nezhad, *Fuzzy job-shop scheduling problems: A review*. *Information Sciences*, 2014. **278**(0): p. 380-407.
- [8] Scheibenpflug, A., et al. *On The Analysis, Classification and Prediction of Metaheuristic Algorithm Behavior for Combinatorial Optimization Problems*. in *Proceedings of the 24th European Modeling and Simulation Symposium, EMSS 2012*. 2012.
- [9] Rahnamayan, S., H.R. Tizhoosh, and M.M.A. Salama. *Quasi-oppositional Differential Evolution*. in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. 2007.
- [10] Chen, Y., et al., *A hybrid grouping genetic algorithm for reviewer group construction problem*. *Expert Systems with Applications*, 2011. **38**(3): p. 2401-2411.
- [11] Yugay, O., et al. *Hybrid Genetic Algorithm for Solving Traveling Salesman Problem with Sorted Population*. in *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*. 2008.
- [12] Maaranen, H., K. Miettinen, and M.M. Mäkelä, *Quasi-random initial population for genetic algorithms*. *Computers & Mathematics with Applications*, 2004. **47**(12): p. 1885-1895.
- [13] Rahnamayan, S., H.R. Tizhoosh, and M.M.A. Salama, *A novel population initialization method for accelerating evolutionary algorithms*. *Computers & Mathematics with Applications*, 2007. **53**(10): p. 1605-1614.
- [14] Kuczapski, A.M., et al., *Efficient generation of near optimal initial populations to enhance genetic algorithms for job-shop scheduling*. *Information Technology and Control*, 2010. **39**(1): p. 32-37.
- [15] Maaranen, H., K. Miettinen, and A. Penttinen, *On initial populations of a genetic algorithm for continuous optimization problems*. *Journal of Global Optimization*, 2007. **37**(3): p. 405-436.
- [16] Puente, J., C.R. Vela, and I.G. Rodríguez. *Fast Local Search for Fuzzy Job Shop Scheduling*. in *ECAI*. 2010.
- [17] Glover, F., *Future paths for integer programming and links to artificial intelligence*. *Computers & Operations Research*, 1986. **13**(5): p. 533-549.
- [18] Blum, C. and A. Roli, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*. *ACM Computing Surveys (CSUR)*, 2003. **35**(3): p. 268-308.
- [19] Behnamian, J., *Survey on fuzzy shop scheduling*. *Fuzzy Optimization and Decision Making*, 2015: p. 1-36.
- [20] Li, J. and Y.-x. Pan, *A hybrid discrete particle swarm optimization algorithm for solving fuzzy job shop scheduling problem*. *The International Journal of Advanced Manufacturing Technology*, 2013. **66**(1-4): p. 583-596.
- [21] Li, J. and Q.-k. Pan, *Chemical-reaction optimization for solving fuzzy job-shop scheduling problem with flexible maintenance activities*. *International Journal of Production Economics*, 2013. **145**(1): p. 4-17.
- [22] Zheng, Y.L. and Y.X. Li, *Artificial bee colony algorithm for fuzzy job shop scheduling*. *International Journal of Computer Applications in Technology*, 2012. **44**(2): p. 124-129.
- [23] Li, J., et al. *Solving fuzzy job-shop scheduling problem by genetic algorithm*. in *2012 24th Chinese Control and Decision Conference (CCDC)*. 2012.
- [24] Liu, J.-j. *Application of optimization genetic algorithm in fuzzy job shop scheduling problem*. in *Intelligent Systems, 2009. GCIS'09. WRI Global Congress on*. 2009. IEEE.
- [25] Gonzalez-Rodriguez, I., C.R. Vela, and J. Puente. *A memetic approach to fuzzy job shop based on expectation model*. in *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*. 2007. IEEE.
- [26] Victor Paul, P., et al., *A new population seeding technique for permutation-coded Genetic Algorithm: Service transfer approach*. *Journal of Computational Science*, 2014. **5**(2): p. 277-297.



- [27] Park, B.J., H.R. Choi, and H.S. Kim, *A hybrid genetic algorithm for the job shop scheduling problems*. Computers & Industrial Engineering, 2003. **45**(4): p. 597-613.
- [28] Ge, H.-W., et al., *An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling*. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 2008. **38**(2): p. 358-368.
- [29] Kuczapski, A.M., et al., *Efficient generation of near optimal initial populations to enhance genetic algorithms for job-shop scheduling*. Information Technology and Control, 2015. **39**(1).
- [30] Fortemps, P., *Job shop scheduling with imprecise durations: a fuzzy approach*. Fuzzy Systems, IEEE Transactions on, 1997. **5**(4): p. 557-569.
- [31] Lin, F.-T., *A job-shop scheduling problem with fuzzy processing times*, in *Computational Science-ICCS 2001*. 2001, Springer. p. 409-418.
- [32] Tavakkoli-Moghaddam, R., N. Safaei, and M. Kah, *Assessing feasible space in a generalized job shop scheduling problem with the fuzzy processing times: a fuzzy-neural approach*. Journal of the Operational Research Society, 2008: p. 431-442.
- [33] Fayad, C. and S. Petrovic, *A Fuzzy Genetic Algorithm for Real-World Job Shop Scheduling*, in *Innovations in Applied Artificial Intelligence*, M. Ali and F. Esposito, Editors. 2005, Springer Berlin Heidelberg. p. 524-533.
- [34] Ghrayeb, O.A., *Solving job-shop scheduling problems with fuzzy durations using genetic algorithms*. 2000, New Mexico State University.
- [35] Ghrayeb, O.A., *A bi-criteria optimization: minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems*. Applied soft computing, 2003. **2**(3): p. 197-210.
- [36] Li, F.-m., et al., *Fuzzy Programming for Multiobjective Fuzzy Job Shop Scheduling with Alternative Machines Through Genetic Algorithms*, in *Advances in Natural Computation*, L. Wang, K. Chen, and Y. Ong, Editors. 2005, Springer Berlin Heidelberg. p. 992-1004.
- [37] González-Rodríguez, I., C. Vela, and J. Puente, *An Evolutionary Approach to Designing and Solving Fuzzy Job-Shop Problems*, in *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, J. Mira and J. Álvarez, Editors. 2005, Springer Berlin Heidelberg. p. 74-83.
- [38] González-Rodríguez, I., C. Vela, and J. Puente, *Study of Objective Functions in Fuzzy Job-Shop Problem*, in *Artificial Intelligence and Soft Computing – ICAISC 2006*, L. Rutkowski, et al., Editors. 2006, Springer Berlin Heidelberg. p. 360-369.
- [39] Gonzalez-Rodriguez, I., et al., *Semantics of Schedules for the Fuzzy Job-Shop Problem*. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 2008. **38**(3): p. 655-666.
- [40] Sakawa, M. and T. Mori, *An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date*. Computers & Industrial Engineering, 1999. **36**(2): p. 325-341.
- [41] Sakawa, M. and R. Kubota, *Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms*. European Journal of Operational Research, 2000. **120**(2): p. 393-407.
- [42] Sakawa, M. and R. Kubota, *Two-objective fuzzy job shop scheduling through genetic algorithm*. Electronics and Communications in Japan (Part III: Fundamental Electronic Science), 2001. **84**(4): p. 60-68.
- [43] Xie, Y., J. Xie, and J. Li, *Fuzzy due dates job shop scheduling problem based on neural network*, in *Advances in Neural Networks–ISNN 2005*. 2005, Springer. p. 782-787.
- [44] Song, X., et al. *A hybrid strategy based on ant colony and taboo search algorithms for fuzzy job shop scheduling*. in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*. 2006. IEEE.
- [45] Bierwirth, C. and D.C. Mattfeld, *Production scheduling and rescheduling with genetic algorithms*. Evolutionary computation, 1999. **7**(1): p. 1-17.
- [46] Palacios, J.J., et al., *Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop*. Fuzzy Sets and Systems, 2015. **278**: p. 81-97.
- [47] Gao, K.Z., et al., *An effective discrete harmony search algorithm for flexible job shop scheduling problem with fuzzy processing time*. International Journal of Production Research, 2015. **53**(19): p. 5896-5911.



- [48] Nguyen, S., *Automatic Design of Dispatching Rules for Job Shop Scheduling with Genetic Programming*. 2013, Victoria University of Wellington.
- [49] Itoh, T. and H. Ishii, *Fuzzy due-date scheduling problem with fuzzy processing time*. International Transactions in Operational Research, 1999. **6**(6): p. 639-647.
- [50] Petrovic, S. and C. Fayad. *A fuzzy shifting bottleneck hybridised with genetic algorithm for real-world job shop scheduling*. in *Proceedings of Mini-EURO Conference, Managing Uncertainty in Decision Support Models, Coimbra, Portugal*. 2004. Citeseer.
- [51] Lei, D., *Pareto archive particle swarm optimization for multi-objective fuzzy job shop scheduling problems*. The International Journal of Advanced Manufacturing Technology, 2008. **37**(1-2): p. 157-165.
- [52] Zhang, G., L. Gao, and Y. Shi, *An effective genetic algorithm for the flexible job-shop scheduling problem*. Expert Systems with Applications, 2011. **38**(4): p. 3563-3573.
- [53] Kyung-Mi, L., T. Yamakawa, and L. Keon-Myung. *A genetic algorithm for general machine scheduling problems*. in *Knowledge-Based Intelligent Electronic Systems, 1998. Proceedings KES '98. 1998 Second International Conference on*. 1998.
- [54] Caumont, A., P. Lacomme, and N. Tchernev, *A memetic algorithm for the job-shop with time-lags*. Computers & Operations Research, 2008. **35**(7): p. 2331-2356.
- [55] González, M.A., C.R. Vela, and R. Varela, *A competent memetic algorithm for complex scheduling*. Natural Computing, 2012. **11**(1): p. 151-160.
- [56] Murphy, G.C., M. Kersten, and L. Findlater, *How are Java software developers using the Eclipse IDE?* Software, IEEE, 2006. **23**(4): p. 76-83.
- [57] Gen, M., Y. Tsujimura, and E. Kubota. *Solving job-shop scheduling problems by genetic algorithm*. in *Systems, Man, and Cybernetics, 1994. Humans, Information and Technology., 1994 IEEE International Conference on*. 1994. IEEE.
- [58] Hu, Y., M. Yin, and X. Li, *A novel objective function for job-shop scheduling problem with fuzzy processing time and fuzzy due date using differential evolution algorithm*. The International Journal of Advanced Manufacturing Technology, 2011. **56**(9-12): p. 1125-1138.
- [59] Moscato, P., *On evolution, search, optimization, GAs and martial arts: toward memetic algorithms*. California Inst. Technol., Pasadena. 1989, CA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826.
- [60] González Rodríguez, I., et al. *A new local search for the job shop problem with uncertain durations*. in *INTERNATIONAL CONFERENCE ON AUTOMATED PLANNING AND SCHEDULING (ICAPS 2008)*. 2008. Sydney: AAAI Press.
- [61] Palacios, J., et al., *Hybrid Tabu Search for Fuzzy Job Shop, in Natural and Artificial Models in Computation and Biology*, J. Ferrández Vicente, et al., Editors. 2013, Springer Berlin Heidelberg. p. 376-385.
- [62] Gao, J., L. Sun, and M. Gen, *A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems*. Computers & Operations Research, 2008. **35**(9): p. 2892-2907.
- [63] Zhang, R. and C. Wu, *A hybrid local search algorithm for scheduling real-world job shops with batch-wise pending due dates*. Engineering Applications of Artificial Intelligence, 2012. **25**(2): p. 209-221.
- [64] Neri, F. and C. Cotta, *Memetic algorithms and memetic computing optimization: A literature review*. Swarm and Evolutionary Computation, 2012. **2**(0): p. 1-14.
- [65] Van Laarhoven, P.J., E.H. Aarts, and J.K. Lenstra, *Job shop scheduling by simulated annealing*. Operations research, 1992. **40**(1): p. 113-125.
- [66] Abdolrazzagah-Nezhad, M. and S. Abdullah, *A Robust Intelligent Construction Procedure for Job-Shop Scheduling*. Information Technology And Control, 2014. **43**(3): p. 217-229.
- [67] Yahyaoui, A., N. Fnaiech, and F. Fnaiech, *A Suitable Initialization Procedure for Speeding a Neural Network Job-Shop Scheduling*. Industrial Electronics, IEEE Transactions on, 2011. **58**(3): p. 1052-1060.
- [68] Li, X. and M. Yin, *An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure*. Advances in Engineering Software, 2013. **55**(0): p. 10-31.
- [69] Kaur, D. and M.M. Murugappan. *Performance enhancement in solving traveling salesman problem using hybrid genetic algorithm*. in *Fuzzy Information Processing Society, 2008. NAFIPS 2008*.



- Annual Meeting of the North American*. 2008. IEEE.
- [70] Victor Paul, P., et al., *Performance analyses over population seeding techniques of the permutation-coded genetic algorithm: An empirical study based on traveling salesman problems*. Applied Soft Computing, 2015. **32**(0): p. 383-402.
- [71] Bahameish, H.A., *Finding a Cost Effective LNG Annual Delivery Program (ADP) Using Genetic Algorithms*. 2014.
- [72] Pallant, J., *SPSS survival manual: A step by step guide to data analysis using SPSS*. 2010: McGraw-Hill International.
- [73] Cohen, J., *Statistical Power Analysis for the Behavioral Sciences*. 2nd ed. 1988, Hillsdale, New Jersey: Erlbaum.