International Journal on Advanced Science Engineering Information Technology

Vol.8 (2018) No. 4-2
ISSN: 2088-5334

Population Initialisation Methods for Fuzzy Job-Shop Scheduling Problems: Issues and Future Trends

Syaimak Abdul Shukor[#*], Iman Mousa Shaheed[#], Salwani Abdullah[#]

[#] *Faculty of Information Sciences and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia*
** E-mail: syaimak@ukm.edu.my*

***Abstract*— Scheduling job shops in the real-world manufacturing environment is a multifarious task that involved various and multiple components, solutions and approach. Fuzzy Job-Shop Scheduling Problems (Fuzzy JSSPs) are most commonly addressed by the population-based Meta-heuristic algorithms. These algorithms usually derive and develop near-optimum results within credible computational times, almost by two main steps; the initialisation and then improvement step. Numerous theoretical studies pointed out that a Meta-heuristic performance is mainly affected by the performance of its initialisation method. The main motivation of this work is to perceive the existing pattern and concerns on population initialisation issues for Fuzzy JSSPs current work by scrutinizing the published articles. Furthermore, this paper focusing on providing comprehension insight and future direction on these methods. Therefore, this paper determined to review and classify the existing literature on Fuzzy JSSPs and analyse the performance of the initialisation methods used to identify their possible limitations. In consequence, previous works outlined three potential methods for initial solutions generation, which are Random-based, priority rules-based, and heuristic methods. However, the current analysis showed that Heuristic-based initialisation approach remains lacking in the Fuzzy JSSPs domain in spite of its successful performance in the crisp JSSP domain, especially, its capability to generate high-quality initial population that consists of optimal or near optimal solutions. Furthermore, this paper identifies probable gaps and reveals several performance limitations in the existing methods, which demands for an urgent solution to develop alternatives. Promising suggestions for future studies are also provided that may lead to new Heuristic Initialisation methods that can be proposed to solve current weaknesses.**

***Keywords*—fuzzy job shop scheduling; population initialisation; scheduling problems; fuzzy JSSP.**

I. Introduction

In the current business environment, the need to increase productivity has driven the manufacturers to exploit every prospect in generating profits and reducing cost [1]. The pressures from the economy and market requirements necessitate minimization of inventory while striving to maintain the customer satisfaction towards the level of quality in the delivery and production. As scheduling is the last step before operation plans are converted into productive activities, it represents a major determinant of production costs and service levels. Poor scheduling leads to the inefficient utilization of resources, increased production costs, reduced competitiveness in the marketplace, and customer dissatisfaction due to delayed orders. Thus, management and production efficiency is heavily dependent on optimal scheduling of the tasks to be performed with limited resources.

In such environment, owing to the complexity of production flows and the uncertainty of the practical requirements, scheduling become a very challenging and vigorous problem [2]- [4]. Therefore, because of its high prospective to radically reduce the costs and increase the throughput, Fuzzy Job Shop Scheduling Problems (Fuzzy JSSPs) are considered as one of the most popular and highly interested research issues in this field.

Since Fuzzy JSSP is the NP-hard problem type, there has been progression of interest regarding improvement and development of meta-heuristic algorithms to solve it, such as; the adoption of population-based meta-heuristic algorithms like memetic algorithm (MA), genetic algorithm (GA), particle swarm optimization (PSO) has led to better results for Fuzzy JSSPs than the traditional dispatching or heuristic algorithm. Most recent literature review by Behnamian [5] shows that Meta-Heuristic algorithms receives major consideration compared to other techniques in the fuzzy scheduling area 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 [6] – [7]. These algorithms commonly derive near-optimum solutions, almost by two main steps [8], which are initialisation step and then improvement step. However, population initialisation considered as a crucial task in population-based meta-heuristic algorithms [6]. As discussed in [9] – [11], the performance of population initialisation claimed to be important compared to any other stage in population-based meta-heuristics that reached the nearest value to the required optimal solution and on the same time has potential to highly escalate the efficiency.

In this context, numerous studies mentioned that the goodness of the initial population can affect meta-heuristics convergence speed. According to [12], population in each iterative improvement process depend on the previous and initial population. Thus, in order to accelerate the computation, initial population acts as a major part in attaining the final optimal solution. This role has also been advocated by [12]- [16] who mentioned that the quality of the initial population, in particular, can accelerates the convergence speed as well as improves finishing solution quality.

Even though there are a lot of effort and attention towards population initialization techniques through substantial number of publications; less attention is given to evaluate and classify them in a methodical and comprehensive approach. At the moment and based on our finding, the only effort in delivering brief assessment of current population initialization techniques for Fuzzy JSSPs was prepared by Abdullah and Abdolrazzagh-Nezhad [8]. This paper endeavours to expand and magnify the previous work in numerous ways and approaches. Foremost, in order to include more recent initialization techniques that was not mentioned by [8], a comprehensive and wide-ranging survey is conducted thoroughly. Next, to gather more insight of the issues related to the research topic, the trends and related open questions are being discussed. Finally, based on discussion and findings, proper and suitable guidelines for the future research are proposed. Indeed, this paper focuses mainly on the prominence, significance and challenges in the population initialization for Fuzzy JSSPs research area. This will assist the researcher in this area to comprehend the complete depiction of the current research and facilitate them in selecting the accurate and suitable population initialization techniques for their future use and research. In real world manufacturing systems there are range of uncertainties linked to (a) customer demands, (b) due and release dates, and (c) processing and transportation represent a spanner in the works for production procedures. These uncertainties could be due to reasons that include an alteration in the customer order, an alteration in the availability date, failure in delivery of raw materials, and engine breakdown.

The Fuzzy JSSP is defined as: There are a set of n jobs to be processed on m machines. Each job i comprises of an order of n_i operations where $O_{i,j}, j = 1, 2, \dots, n_i$. Each routing has to decided in order to finsih all the processes of a job. The implementation and execution of $O_{i,j}$ needs 1 machine from a set of given machines $M_i \subseteq M$. The processing time of the $O_{i,j}$ on the particular selected machine *k* is characterized as a Two Fuzzy Numbers (TFN) $t_{i,j,k} = (t^1_{i,j,k}, t^2_{i,j,k}, t^3_{i,j,k})$, Equally, the fuzzy completion time of job $O_{i,j}$ is denoted as a TFN $C_{i,j} = (C^1_{i,j}, C^2_{i,j}, C^3_{i,j})$, where $C^1_{i,j}$ is defined as the shortest completion time, while $C^2_{i,j}$ is expressed as the most probable completion time and finally $C^3_{i,j}$ is knows as the largest completion time. The allocation of machines and the order of the operations is determined by Fuzzy JSSP for all machines involved in order to reduce particular and certain interest or objective, as example, the maximum fuzzy completion time is defined as:

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

where C_i defines as fuzzy completion time of job *i*.

In order to produce a schedule, some fuzzy number operations are required and essential in fuzzy context. The operations involved are the ranking technique or method of fuzzy numbers, the addition operation, the max operation of two fuzzy numbers. For calculation of fuzzy completion time of an operation, additional operations is usually adopted. To determine the fuzzy beginning time of an operation, Max operation is implemented. While ranking technique or method is usually used in comparing the maximum fuzzy completion time. The following formula shows the addition of two triangular fuzzy numbers (TFNs), $A = (a^1, a^2, a^3)$ and $B = (b^1, b^2, b^3)$:

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

While developing a feasible and practicle schedule, in order to rank fuzzy numbers; the following criterion is implemented.

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

Step2: If both TFNs own the indentical $C_1(A)$, this will results in $C_2(A) = a^2$ will be selected as the second criterion.

Step3: If $C_1$and C_2 are identical, this will make $C_3 = a^3 - a^1$ selected as the third criterion. In fuzzy scheduling , the above criterion was demonstrated to deliver better performance. Therefore, the max of two TFNs A and B usually estimated by this criterion. That is, if $A > B$, then $A \vee B = A$ else $A \vee B = B$.

II. Material and Method

Population initialization techniques consist of several types of diverse features and characteristics. Currently, two types of initialization methods are recognized in the Fuzzy JSSPs literature, which are Random–based and priority rules-based methods. According to [8], the most common and preferred population generation technique is the Random Population Initialization which is straightforward and simple when preceding information on the problem to solve them are lacking. Existing works reveal that random techniques are the preferred choice of most researchers for generating the initial population. This is may due to these methods easy execution, sufficient divergence of the initially produced points in the problem of space searching and extremely brief calculation period. Below are review to the selected papers that used random-based methods for the Fuzzy JSSPs initialisation.In the research by Fortemps [17], the initial solution is obtained as the following: each task were considered consecutively in random order and operations were put on the list based on the suitable or matching machine, the sequence will always apprently considered as very poor. The conversion into an element of the solution space is a direct process. Then, the coordination of some critical arcs of the disjunctive graph were degenerated in order to progress to another feasible orders.

As mentioned in [18], Johnson's constructive algorithm has been used to generate solutions. However, this algorithms randomly orders the jobs. A fuzzy-neural method of constraint satisfaction for a generalized job shop scheduling problem (GJSSP) fuzzy processing times is initiated in [19]. It proposed the following steps: (a) The processing time of operations as a triangular fuzzy number are possessed. (b) Each operation is allocated to any active machine. (c) W_{ST} , $W_{PT}, \varepsilon, \delta$ and t=0 are set (d) $S_{ikp}(O) = U(O, S_{Max}) \forall i, k, p, O_{ikp} \in R_p$ is set where randomly $U(a, b)$ represents the continuous uniform distribution and S_{Max} defines as a large random number and lastly (e) $T_{ikp}(0) = M_{ikp} \forall i, k, p, O_{ikp} \in R_p$ is finally set. Triangular fuzzy numbers are usually implemented to denote the job shop scheduling problems with uncertain durations and furthermore random numbers are considered in generating initial solutions for the experiments [20] – [37]. However, this procedure may produce illegal solutions. In order to avoid construction of illegal chromosome, the N chromosomes were stochastically formed as the initial population. Each chromosome is made up of n×m natural number codes in [1,n] [38]- [41] with N represent the population number. Random number were used to yield natural number in [1,n] representation and record times of the number. However, to generate feasible fuzzy schedules, [42]-[48] have extended the G&T algorithm, which were then implanted with random selections to produce initial fuzzy schedules. Subsequent to this, the fuzzy schedules were implanted with random selections to yield initial fuzzy schedules. This is to come up with viable fuzzy schedules. Song et. al [49] used the extension of G&T algorithm by Bierwirth and Mattfeld (1999) to form the initial swarm.

Palacois et. al [50] suggested a random initialisation method based on the G&T algorithm. The following steps or phases generated initial solutions; operations were scheduled in an insertion mode by manipulating the flexibility, a reasonable insertion interval for an operation expressed by o_{ij} in a machine $\mathrm{M}_k \in \mathrm{M}(\mathrm{o}_{i,j})$ to be a dedicated time interval $[\mathrm{t}^S, \mathrm{t}^E]$, where machine M_k is in idle status and for that o_{ij} can be processed within that particular time interval without violating the preference constraints. For initialising population, Gao et. al [51] has recommended *MinEnd* heuristic which is a new heuristic approach. Here, the decision on the operation order or sequence is randomly determined. For each operation, the assignment of the processing machine is based on the operation sequence or order. The following are the heuristic steps adopted; Step 1: To obtain the operation sequence (OS), all jobs operations are randomly shuffled. Step 2: Repair OS, ensure that the processing priority is satisfied through the same job operations. Step 3: In OS, for each operation $(O_{i,j})$, fuzzy completion time is assessed on each selectable machine. Step 4: For processing operation $O_{i,j}$, the machine that owns minimum fuzzy completion time is selected. On the other side, priority rules-based initialisation methods have been used in many studies. Itoh and Ishii [52] used an extended due-date rule that Ordering the objective jobs according to their due date. Kuroda and Wang [53] used a kind of slack priority rule considering the fuzziness of the due date and processing times. Petrovic and Fayad [54] 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 [38] altered the process and one of four rules was selected on random basis for sub-chromosome handling: EDD, SPT, LPT, and LRT. Deming Lei [55] 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 negated the potential incorrect representation, caused by the usage of a random range. The initial population in the algorithms through the merging of a set of priority rules and random selections was generated by [56]- [57]. Nalepa et. al [58] considered different population initialization strategies including random method and set of priority rules such as (First), which the operations are allocated to the first available machine. (Last), which allocate operations to the last machine, as well as tactics which maximize the utilization of the machines (max BU). Furthermore, (min BU) represents item that targets in generating the even spread of operations for all available machines.

In this section, the characterisations of the existing initialisation methods for Fuzzy JSSPs were discussed. The summary of these strength and weaknesses is presented in Table 1.

The main objective is to analyse and perceive the existing trend, issues and complications of population initialisation for Fuzzy JSSPs research. This is done by exploring and investigating the related published articles. Furthermore, it is the aim of this study to suggest and provide practitioners, researchers and academics with vision, understanding and future direction on these methods. Therefore, year of publication of the selected research papers of Fuzzy JSSPs were adopt to verify the distribution of the papers. In order to classify the selected research papers, two methods were selected in this study: the techniques used for initialisation and the criteria applied to evaluate their performance. Hence, several major electronic databases were explored and examined in order to provide a thorough and inclusive bibliography of research papers related to initialisation methods. Among the electronic databases searched are:

- ACM Portal;
- Wiley Online Library;
- IEEE Xplore;
- Taylor & Francis;
- Science Direct;
- Google Scholar;
- Springer.

TABLE I
THE STRENGTH AND DRAWBACKS OF THE EXISTING INITIALISATION METHODS

Method	Advantage	Disadvantage
Random-based	- Easy to execute. - Extremely brief calculation period. - Sufficient diversification of the initially produced points in the problem search area.	- Requires longer and lengthier time in calculating the optimal solution for Fuzzy JSSP. - Random population could decrease the opportunity of attaining an optimal solution. - Lack of useful diversity.
Priority rules-based	- Easily implemented. - Less demanding when it comes to calculations - They are not bogged down by time intricacy.	- Restricted to the generation of a single solution. - Decrease the population diversity - Losses wide areas or regions of the search space.

The first phase of publications search was performed based on the following search criteria: “fuzzy job shop” and "metaheuristics", “job shop” and “uncertainty” and “metaheuristics” in any of the abstracts, keywords or titles. For the first phase, the searching process could not detect the papers with exact terms of the metaheuristics techniques. Therefore, in the second phase the specific and exact term of metaheuristic techniques were searched. Each full text paper was reviewed, and any paper that did not declared or mentioned any initialisation methods was discarded. Finally, 41 research papers on Fuzzy JSSPs were selected from 28 journals. Next, each paper was thoroughly analysed and reviewed. In order to view the literature from various perspectives, the papers were classified based on the classification framework stated in the next section. Even though the exploration and investigation was not comprehensive, this analysis provides exhaustive and in-depth foundation in understanding and appreciating the research related to population initialisation. The classification framework proposed for this work comprises of initialisation techniques, performance factors and benchmark datasets characteristics. Here, the selected papers were classified into three classifications of techniques and seven categories of performance factors. Figure 1 illustrates the proposed classification framework for Fuzzy JSSPs graphically.

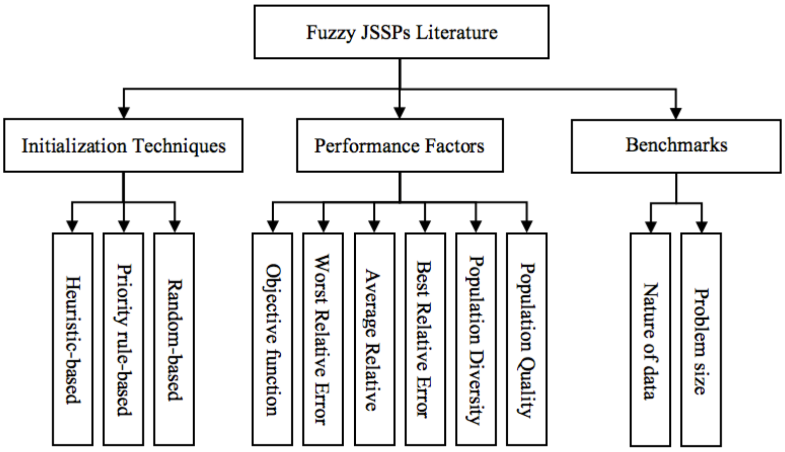

Fig.1. Classification framework

A. Initialisation Techniques

In order to employ any Meta-Heuristic algorithm to address the Fuzzy JSSPs there are mainly two main required steps. The initialisation and then improvement step. Notably, Meta-Heuristics performance is mainly affected by the performance of its initialisation method, by means the quality and the diversity of the initial solutions generated in this step ([6], [59] - [60]). Previous works outlined three methods for initial JSSPs solution generation. There are Random techniques, priority rules, and Heuristic approaches [61].

i) *Random-based initialisation*: random initialisations generate initial population in arbitrary way. These methods are more often used [42] - [44]. This may due to simple and straightforward implementation, and also produce very short computational time.

ii) *Priority rules-based initialisation*: Priority rule is a simple heuristic implemented in a scheduling problem for tasks sequencing. Now when decisions were required for sequencing, the jobs in the queue of the machine will be ranked and determined by priority rules. Then, the job that owns the highest priority is next selected to be processed at particular corresponding machine [62]. According to Abdullah and Abdolrazzagh-Nezhad [61], priority rules came second after random techniques in its usage for Fuzzy JSSPs initial population generation. This popularity can be traced to the fact that they are easily implemented, are less demanding when it comes to calculations, and are not bogged down by time intricacy.

iii) *Heuristic-based initialisation*: heuristics are techniques that rely on experience in achieving good solutions to computational problems. Usually, Heuristics are developed for the management of situations where the exact methods for solving a computational problem are impractical (e.g. too time-consuming) [63].

B. *Performance factors*

In this section several measures are described which can be used to analyse the performance of population initialisation techniques for meta-heuristic algorithms when solving combinatorial optimization problems such as Fuzzy JSSPs:

i) *Fuzzy Objective function*: The main five objective functions are described as:

- *Maximising the satisfaction level:* The objective function was introduced by [48] and used for Fuzzy JSSPs. What is meant by satisfaction level is the fuzzy due date's membership function. It relates to the level of satisfaction the decision maker reaches depending on the job's completion time. The objective's main purpose is to monitor for fuzzy due dates' severe breaches and delays.
- *Optimum minimisation of the de-fuzzed makespan:* [64]- [65] were the first sources to present this Fuzzy JSSPs objective function, stressing on the importance of fuzzy processing time. Through the fuzzy number's mean value usage, the researchers de-fuzzed the fuzzy processing time and substituted the fuzzy values through corresponding crisp de-fuzzed processing times, with the purpose in reaching the research's objective. In short, the goal was to minimise the de-fuzzed makespan pertaining to the crisp de-fuzzed JSSP.
- *Minimise the fuzzy makespan's uncertainty and total integral value:* Niu et. al [32] were the one to first apply the particular objective function in the purpose of Fuzzy JSSPs optimisation. The unclear criteria with two crisp criteria is replaced. These devised from two crisp values assigned on the fuzzy makespan. The fuzzy makespan's whole integral value represents a convex derived by both left and right integral value using an optimism index. The left and right integrals determine whether the fuzzy makespan is pessimistic or optimistic, respectively. The fuzzy makespan's spread is also a way to reduce its vagueness.
- *Minimising the likely fuzzy makespan:* The target of this objective function is optimising the Fuzzy JSSPs' fuzzy processing time. As the projected value of a triangular fuzzy number which is ($\mathrm{E}[\tilde{A}]$) equals to the neutral scalar substitute of a fuzzy interval. Furthermore, the centre of gravity of its mean value can be attained. As a series of researchers ([33], [50], [56]-[57]) considered this criterion's detraction of fuzzy makespan ($\min \mathrm{E}[\tilde{C}_{\max}]$). Current study acknowledges this objective function as a crucial decision, which needs to be reached.
- *Maximise the agreement index's minimum:* ([23], [66]) posit that this function is focused on the optimisation of the fuzzy completion time part which is fulfilled as a common Fuzzy JSSPs criterion by the fuzzy due date, utilising both fuzzy processing time and also fuzzy due date. In order to gauge customer's contentment of the job's time for completion, the agreement index is best described as the value of the fuzzy circumstance, a degree of either earliness or delay,

ii) *The Error Rates*: In Fuzzy JSSPs, for measuring the initialisation procedures performance, with an objective function of reducing the likely fuzzy makespan to the lowest possible level, the fitness of the best, average and worst solutions produced by each initialisation method have been calculated and compared in numerous studies ([11], [60], [67]). Therefore, this factor will be considered in this study. However, in this study the best relative error (BRE), the average relative error (ARE), and the worst relative error (WRE) will be calculated as in [56]:

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

$$\mathrm{ARE}=\Big(\sum_{i=1}^{n}\frac{\mathrm{Sol_i}\text{-}\mathrm{Best_{known}}}{\mathrm{Best_{known}}}\times 100\Big)/n \qquad (4)$$

$$\mathrm{WRE}=\frac{\mathrm{Sol_{worst}}\text{-}\mathrm{Best_{known}}}{\mathrm{Best_{known}}}\times 100 \qquad (5)$$

where, $Best_{known}$ is the best-known solution in the literature. *n* is the population size. Sol_{best}, Sol_{worst} are the best and worst solution by the initialisation method respectively. Sol_i is the i^{th}solution by the specific initialisation method. BRE, ARE, and WRE can be used to answer the question of how close the best, worst solutions and the average fitness of the produced population from the best known solution so far. The smallest value of BRE means that the best solution is close to the optimal solution so far, for a specific problem instance.

iii) *The Population Quality*: For measuring the initial population quality, the generated average of the convergence rate of solutions in the initial population will be used [68]. Victer Paul et al. [69] did use it to measure initial population quality. It can be given as:

$$\text{Average convergence}(\%) = \Big(1\text{-}\frac{\text{Average fitness-}\mathrm{Best_{known}}}{\mathrm{Best_{known}}}\Big)\times 100 \qquad (6)$$

Average fitness is define as the average fitness value for the solutions in the population; while $\mathrm{Best_{known}}$ is described as the best recognized solution for the corresponding instance. Generated population quality can be measured using this factor as average of the population. The performance of convergence time can be increased through initial population paired with good average convergence. It also provides better assistant in exploring the search space [50].

iv) *The Population Diversity*: In this study, the phenotypic diversity that proposed by [70] is used to measure population diversity. In this type of diversity measurement, the population diversity can be obtained by calculating the difference between individuals. For the Fuzzy JSSP; the combinatorial entity is a schedule which distinctively defined by the arrangement jobs location on each machine. Therefore, the number of jobs that are differently arranged on each machine can be considered as the usual approach to check level of differences on two given schedules. Subsequently, Brizuela and Sannomiya [70] proposed the following definition:

$$\text{dif}(a,b)=\sum_{i=1}^{m}\sum_{j=1}^{n}\partial_{ab}(\mathbf{i},j \qquad (7)$$

Where $\mathbf{dif}(a,b)$ is defined as difference between two Fuzzy JSSP schedules named as a and b, with m machines and n jobs, is known by the sum over all machines of the number of differently sequenced jobs on each machine. Where

$$\partial_{ab}(i,j)=\begin{cases}0,\ \text{if job } j \text{ in machine } i \text{ of schedule } a\ (\text{i.e. } (j,i,a)) \\ \text{is immediately} \\ \text{followed by the same job that follows } (j,i,b) \\ 1,\ \text{otherwise}\end{cases} \qquad (8)$$

The population diversity of G individuals for a JSSP with *n* jobs and *m* machines with difference $dif(a,b)$ between schedules a and b is defined as:

$$\text{Div}=\frac{2}{mnG(G-1)}\sum_{i=1}^{G-1}\sum_{k=1}^{G-i}\text{dif}(i,i+k) \qquad (9)$$

v) *The Computational Time*: The CPU time is calculated in JAVA using the function System.nanoTime(), which returns the running time in Nano seconds and subsequently converted to seconds. The function is used right before creating the first generation and called again after completing each subsequent generation [71].

C. Benchmarks

i) *Size of problem*: in this classification, the three parts used by [72] were adopted. These are small-scale datasets that range from 100 (10 jobs, 10 machines) to 375 operations, medium-scale datasets range from 400 to 1,500 operations, and large-scale datasets range from 2000 to 3,750 operations.

ii) *Nature of data*: this classification distributes datasets into two types, (1) scholar-generated datasets and (2) real-world datasets.

III. Result and Discussion

From the selected papers from 28 publishers, they were classified based on the proposed classification framework. From the thorough analysis of the selected papers, the outcomes will provide guidelines and recommendations for future exploration and research on initialisation methods. Below are further discussions on this particular topic on initialisation methods.

A. Distribution by initialisation technique

Figure 2 indicates the distribution of selected research papers by method. It can be seen that 84% of the selected research papers which can be considered as majority were related to random method. While, priority rules-based methods came second with 16.0%. However, it can be observed that there is no heuristic-based initialisation method has developed for Fuzzy JSSPs.

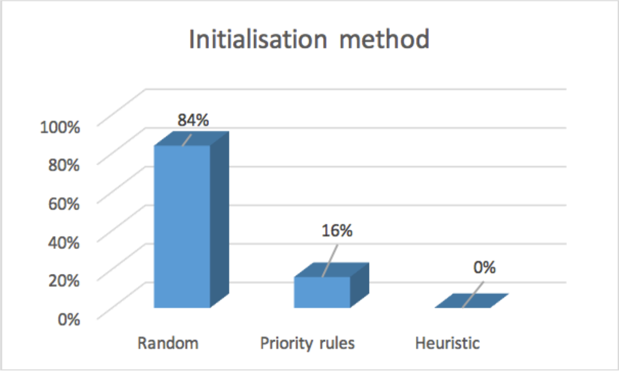

Fig.2. The population initialisation for Fuzzy JSSPs

B. Distribution by performance factors

Figure 3 shows the distribution of research papers by factors used to evaluate the performance. It can be reported that majority of the papers which is 39.53% used the objective function as a performance measure. The second rank performance method used was the convergence time (27.91%). The average relative error (ARE) are used in 18.6% of the research papers. While WRE used in 11.63% of the studies. A little attention was paid to the BRE that measured in only 2.33%. Unfortunately, it can be observed that the population quality and diversity have not been measured in all studies reviewed. However, Palacios, et al. [50] seems to be the only paper that measured the CPU time, best and average makespan for the initial population while other studies tend to measure the population after improvement.

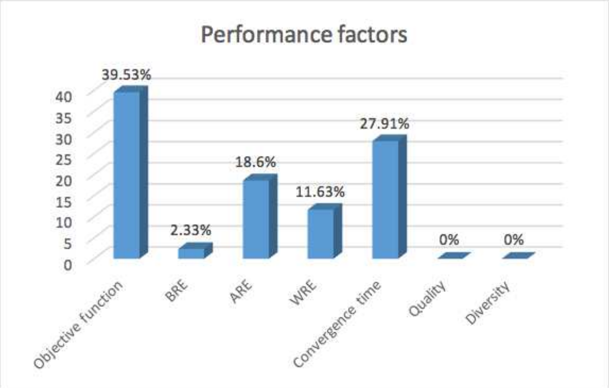

Fig.3. Distribution by performance factors

C. Distribution by Fuzzy JSSPs Benchmarks

a) Problem size

Distribution of research papers by problem size of the Fuzzy JSSP benchmark datasets is represented in Figure 4. The majority of studies used small-scale datasets (98.0%). However, only 2.0% of the studies reviewed in this paper has used medium-scale datasets in their experiments. Large-scale datasets have never been tested yet.

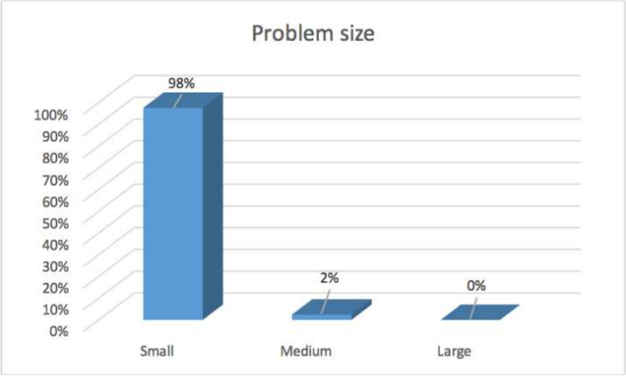

Fig.4. Distribution of Fuzzy JSSPs Benchmark datasets by problem size

b) Nature of data

Figure 5 represents the distribution of selected papers based on the nature of data of the Fuzzy JSSP benchmark datasets. The majority of studies used Scholar-generated datasets (95.0%). While only 5.0% of the studies reviewed in this paper has used Real-world datasets in their experiments.

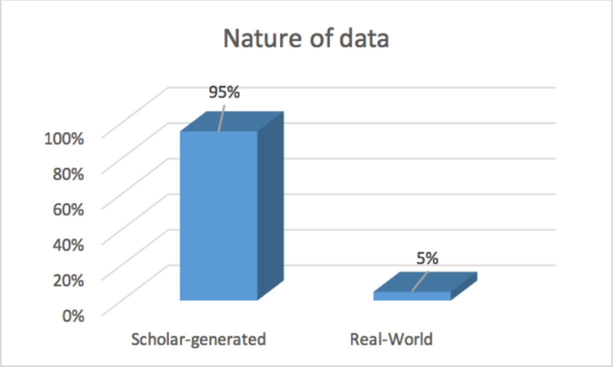

Fig.5. Distribution of Fuzzy JSSPs benchmark datasets by Nature of data

IV. Conclusions

Addressing and discussing on Fuzzy JSSPs by meta-heuristic algorithms have appealed and attracted the focus of the practitioners, researchers and academics in this field. In order to understand and appreciate the development and progress of Fuzzy JSSPs initialization research and furthermore to provide them with understanding, insight and forthcoming direction of the Fuzzy JSSPs initialization methods; 41 research papers between 1996 to 2018 on Fuzzy JSSPs were identified.

The outcomes of the analysis in this work proposed some substantial suggestions and implications towards the research in this field: Most research on Meta-Heuristics applications when it comes to Fuzzy JSSP problems focuses on random methods for developing initial population. The randomness of these techniques, however, increases the necessary calculation time needed to reach a capital Fuzzy JSSP resolution. Additionally, satisfactory solution cannot always be reached, which may lead to illegitimate generation of encoded points by given encoding devices. Unfortunately, the development of advanced population initialisation, such as heuristic-based method is given little attention by the current researches.

However, this relationship has never been investigated quantitatively. Therefore, a quantitative study to establish this correlation could be one factor to encourage the researchers pay attention to the importance of the initialisation process.

Although, theoretically the final solution quality and convergence speed can be influenced by the goodness of the initial population in terms of its quality and diversity, the evaluation of initial population quality and diversity have totally overlooked. Therefore, researchers are motivated to do so in the future.

Most studies used small datasets in their evaluation process. The effectiveness of the developed methodologies has rarely tested on medium-scale dataset while never examined on large ones. Therefore, it is recommended to employ large and more complex datasets in future studies.

There is an obvious absence to the real-world data to be used in the evaluation of the proposed methodologies. This is due to the lack of available cooperative network between the industries and the researchers. Therefore, developing tools that can ease real data collection, especially, online ones are most suggested for future development.

Despite of Heuristic initialisation approaches can develop an initial population characterised by a nearly optimum solution. Therefore, researches are encouraged to pay more attention on developing a Heuristic initialisation approach for Fuzzy JSSPs in reducing the computational time of the improvement algorithm.

Unfortunately, heuristic procedures that already exist are characterised by complicated structures and their time for calculation is extended compared to priority rules processes and random techniques. Furthermore, the existing Heuristic initialisation approach focuses on a very tiny section of the search space which obstructs the generation of an optimal solution because of the initial population's scant assortment. This could be why the heuristic initialisation method is still lacking in the literature of Fuzzy JSSPs. Therefore, the researcher is still about to witness the application of new Heuristic initialisation approaches in the Fuzzy JSSP area to enhance the initial population quality and diversity in reducing the computational time of the improvement algorithm.

Reverting to the weaknesses of existing initialisation methods summarised in Table 1 along with the abovementioned observations, the conclusion can be established accordingly, and therein initial populations generated by existing initialisation methods are lacking of initial population with a good quality and diversity, which could increase the computational time of the improvement algorithms to solve Fuzzy JSSPs. Contrariwise, the promising performance and future development of Heuristic initialisations in providing a good quality initial population may lead the researchers to propose new Heuristic initialisation methods that can overcome the existing shortcomings.

Acknowledgment

We thank Universiti Kebangsaan Malaysia for the support under GGP-2017-078 (Geran Galakan Penyelidikan).

References

[1] Roshanaei, V. (2012). *Mathematical Modelling and Optimization of Flexible Job Shops Scheduling Problem.* (M.A.Sc.), University of Windsor.

[2] Zalmiyah Zakaria, Safaai Deris, Muhamad Razib Othman, Shahreen Kasim (2017). Non-Reshuffle-Based Approach for Rescheduling of Flexible Manufacturing System. *International Journal on Advanced Science Engineering, Information Technology, 4(2),* 1543-1552.

[3] Kuroda, M., & Wang, Z. (1996). Fuzzy job shop scheduling. *International Journal of Production Economics, 44*(1–2), 45-51.

[4] Maroosi, A., Muniyandi, R.C., Sundararajan, E., Md Zain, A., (2016). A Parallel Membrane Inspired Harmony Search for Optimization Problems: A Case Study based on a Flexible Job Shop Scheduling Problem. *Applied Soft Computing,* 49 120-136.

[5] Behnamian, J. (2015). Survey on fuzzy shop scheduling. *Fuzzy Optimization and Decision Making*, 1-36.

[6] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2007, 25-28 Sept. 2007). *Quasi-oppositional Differential Evolution.* Evolutionary Computation, 2007.

[7] Victer Paul, P., Ramalingam, A., Baskaran, R., Dhavachelvan, P., Vivekanandan, K., & Subramanian, R. (2014). A new population seeding technique for permutation-coded Genetic Algorithm: Service transfer approach. *Journal of Computational Science, 5*(2), 277-297.

[8] Abdolrazzagh-Nezhad, M., & Abdullah, S. (2014). A Robust Intelligent Construction Procedure for Job-Shop Scheduling. *Information Technology and Control, 43*(3), 217-229.

[9] Chen, Y., Fan, Z.-P., Ma, J., & Zeng, S. (2011). A hybrid grouping genetic algorithm for reviewer group construction problem. *Expert Systems with Applications, 38*(3), 2401-2411.

[10] Yugay, O., Kim, I., Kim, B., & Ko, F. I. S. (2008, 11-13 Nov. 2008). *Hybrid Genetic Algorithm for Solving Traveling Salesman Problem with Sorted Population.* Paper presented at the Convergence and Hybrid Information Technology, 2008. ICCIT '08.

[11] Abdolrazzagh-Nezhad, M., & Abdullah, S. (2017), Job Shop Scheduling: Classification, Constraints and Objective Functions. *International Journal of Computer and Information Engineering.* 11(4), 429-434.

[12] Maaranen, H., Miettinen, K., & Mäkelä, M. M. (2004). Quasi-random initial population for genetic algorithms. *Computers & Mathematics with Applications, 47*(12), 1885-1895.

[13] Ge, H.-W., Sun, L., Liang, Y.-C., & Qian, F. (2008). 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, 38*(2), 358-368.

[14] Park, B. J., Choi, H. R., & Kim, H. S. (2003). A hybrid genetic algorithm for the job shop scheduling problems. *Computers & Industrial Engineering, 45*(4), 597-613.

[15] Zhang, G., Gao, L., & Shi, Y. (2011). An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Systems with Applications, 38*(4), 3563-3573.

[16] Moaath Shatnawi, Mohammad Faidzul Nasrudin, Shahnorbanun Sahran, (2017). A New Initialization Technique in Polar Coordinates for Particle Swarm Optimization and Polar PSO. *International Journal on Advanced Science Engineering, Information Technology, 7(1),* 242-249.

[17] Fortemps, P. (1997). Job shop scheduling with imprecise durations: a fuzzy approach. *Fuzzy Systems, IEEE Transactions on, 5*(4), 557-569.

[18] Lin, F.-T. (2001). A job-shop scheduling problem with fuzzy processing times. In *Computational Science-ICCS 2001* (pp. 409-418): Springer.

[19] Tavakkoli-Moghaddam, R., Safaei, N., & Kah, M. (2008). Accessing feasible space in a generalized job shop scheduling problem with the fuzzy processing times: a fuzzy-neural approach. *Journal of the Operational Research Society,* 431-442.

[20] García-Álvarez, J., González-Rodríguez, I., Vela, C. R., González, M. A., & Afsar, S. (2018). Genetic fuzzy schedules for charging electric vehicles. *Computers & Industrial Engineering, 121,* 51-61.

[21] González Rodrıguez, I., Vela, C. R., Hernández-Arauzo, A., & Puente, J. (2009). *Improved local search for job shop scheduling with uncertain durations.* Paper presented at the Proc. of ICAPS.

[22] Gonzalez-Rodriguez, I., Puente, J., Vela, C. R., & Varela, R. (2008). Semantics of Schedules for the Fuzzy Job-Shop Problem. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 38*(3), 655-666.

[23] Hu, Y., Yin, M., & Li, X. (2011). 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, 56*(9-12), 1125-1138.

[24] Lei, D. (2009a). A genetic algorithm for flexible job shop scheduling with fuzzy processing time. *International Journal of Production Research, 48*(10), 2995-3013.

[25] Lei, D. (2009b). Genetic Algorithm for Job Shop Scheduling under Uncertainty. In U. Chakraborty (Ed.), *Computational Intelligence in Flow Shop and Job Shop Scheduling* (Vol. 230, pp. 191-228): Springer Berlin Heidelberg.

[26] Lei, D. (2010a). Fuzzy job shop scheduling problem with availability constraints. *Computers & Industrial Engineering, 58*(4), 610-617.

[27] Lei, D. (2010b). Solving fuzzy job shop scheduling problems using random key genetic algorithm. *The International Journal of Advanced Manufacturing Technology, 49*(1-4), 253-262.

[28] Lei, D. (2011). Scheduling fuzzy job shop with preventive maintenance through swarm-based neighbourhood search. *The International Journal of Advanced Manufacturing Technology, 54*(9-12), 1121-1128.

[29] Lei, D. (2012). Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling. *Applied Soft Computing, 12*(8), 2237-2245.

[30] Lei, D., & Guo, X. (2011). Swarm-based neighbourhood search algorithm for fuzzy flexible job shop scheduling. *International Journal of Production Research, 50*(6), 1639-1649.

[31] Li, J., Pan, Q.-K., Suganthan, P. N., & Tasgetiren, M. F. (2012). Solving fuzzy job-shop scheduling problem by a hybrid PSO algorithm. In *Swarm and Evolutionary Computation* (pp. 275-282): Springer.

[32] Niu, Q., Jiao, B., & Gu, X. (2008). Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time. *Applied Mathematics and Computation, 205*(1), 148-158.

[33] Puente, J., Vela, C. R., Hernández-Arauzo, A., & González-Rodríguez, I. (2010). Improving local search for the fuzzy job shop using a lower bound. In *Current Topics in Artificial Intelligence* (pp. 222-232): Springer.

[34] Thammano, A., & Teekeng, W. (2015). A modified genetic algorithm with fuzzy roulette wheel selection for job-shop scheduling problems. *International Journal of General Systems, 44*(4), 499-518.

[35] Wang, S., Wang, L., Xu, Y., & Liu, M. (2013). An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *International Journal of Production Research, 51*(12), 3778-3793.

[36] Xu, Z., Gu, X., Jiao, B., & Gu, J. (2009). *Research on job shop scheduling under uncertainty.* Paper presented at the Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation.

[37] Zheng, Y.-l., Li, Y.-x., & Lei, D.-m. (2011). Multi-objective swarm-based neighbourhood search for fuzzy flexible job shop scheduling. *The International Journal of Advanced Manufacturing Technology, 60*(9), 1063-1069.

[38] Fayad, C., & Petrovic, S. (2005). A Fuzzy Genetic Algorithm for Real-World Job Shop Scheduling. In M. Ali & F. Esposito (Eds.), *Innovations in Applied Artificial Intelligence* (Vol. 3533, pp. 524-533): Springer Berlin Heidelberg.

[39] Ghrayeb, O. A. (2000). *Solving job-shop scheduling problems with fuzzy durations using genetic algorithms.* New Mexico State University,

[40] Ghrayeb, O. A. (2003). A bi-criteria optimization: minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems. *Applied Soft Computing, 2*(3), 197-210.

[41] Liu, J.-j. (2009). *Application of optimization genetic algorithm in fuzzy job shop scheduling problem.* Paper presented at the Intelligent Systems, 2009. GCIS'09. WRI Global Congress.

[42] González Rodrıguez, I., Vela, C. R., Puente, J., & Varela, R. (2008). *A new local search for the job shop problem with uncertain durations.* International conference on automated planning and scheduling (ICAPS 2008), Sydney.

[43] González-Rodríguez, I., Vela, C., & Puente, J. (2005). An Evolutionary Approach to Designing and Solving Fuzzy Job-Shop Problems. In J. Mira & J. Álvarez (Eds.), *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach* (Vol. 3562, pp. 74-83): Springer Berlin Heidelberg.

[44] González-Rodríguez, I., Vela, C., & Puente, J. (2006). Study of Objective Functions in Fuzzy Job-Shop Problem. In L. Rutkowski, R. Tadeusiewicz, L. Zadeh, & J. Żurada (Eds.), *Artificial Intelligence and Soft Computing – ICAISC 2006* (Vol. 4029, pp. 360-369): Springer Berlin Heidelberg.

[45] Li, F.-m., Zhu, Y.-l., Yin, C.-w., & Song, X.-y. (2005). Fuzzy Programming for Multiobjective Fuzzy Job Shop Scheduling with Alternative Machines Through Genetic Algorithms. In L. Wang, K. Chen, & Y. Ong (Eds.), *Advances in Natural Computation* (Vol. 3611, pp. 992-1004): Springer Berlin Heidelberg.

[46] Sakawa, M., & Kubota, R. (2000). Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. *European Journal of Operational Research, 120*(2), 393-407.

[47] Sakawa, M., & Mori, T. (1999). An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date. *Computers & Industrial Engineering, 36*(2), 325-341.

[48] Xie, Y., Xie, J., & Li, J. (2005). Fuzzy due dates job shop scheduling problem based on neural network. In *Advances in Neural Networks–ISNN 2005* (pp. 782-787): Springer.

[49] Song, X., Zhu, Y., Yin, C., & Li, F. (2006). *A hybrid strategy based on ant colony and taboo search algorithms for fuzzy job shop scheduling.* Paper presented at the Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress.

[50] Palacios, J. J., González-Rodríguez, I., Vela, C. R., & Puente, J. (2015). Co-evolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop. *Fuzzy Sets and Systems, 278*, 81-97.

[51] Gao, K. Z., Suganthan, P. N., Pan, Q. K., & Tasgetiren, M. F. (2015). An effective discrete harmony search algorithm for flexible job shop scheduling problem with fuzzy processing time. *International Journal of Production Research, 53*(19), 5896-5911.

[52] Itoh, T., & Ishii, H. (1999). Fuzzy due-date scheduling problem with fuzzy processing time. *International Transactions in Operational Research, 6*(6), 639-647.

[53] Kuroda, M., & Wang, Z. (1996). Fuzzy job shop scheduling. *International Journal of Production Economics, 44*(1–2), 45-51.

[54] Petrovic, S., & Fayad, C. (2004). *A fuzzy shifting bottleneck hybridised with genetic algorithm for real-world job shop scheduling.* Paper presented at the Proceedings of Mini-EURO Conference, Managing Uncertainty in Decision Support Models, Coimbra, Portugal.

[55] Lei, D. (2008). Pareto archive particle swarm optimization for multi-objective fuzzy job shop scheduling problems. *The International Journal of Advanced Manufacturing Technology, 37*(1-2), 157-165.

[56] Li, J., & Pan, Y.-x. (2013). A hybrid discrete particle swarm optimization algorithm for solving fuzzy job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology, 66*(1-4), 583-596.

[57] Li, X., & Yin, M. (2013). An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure. *Advances in Engineering Software, 55*(0), 10-31.

[58] Nalepa, J., Cwiek, M., & Zak, L. (2017). *Behind the Scenes of Deadline24: A Memetic Algorithm for the Modified Job Shop Scheduling Problem.* Paper presented at the International Conference on Man–Machine Interactions.

[59] Shaheed, I.M., Shukor, S.A., Nababan, E.B. (2016). 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. *Journal of Theoretical and Applied Information Technology. 93(2),* 297-311.

[60] Kuczapski, A. M., Micea, M. V., Maniu, L. A., & Cretu, V. I. (2010). Efficient generation of near optimal initial populations to enhance genetic algorithms for job-shop scheduling. *Information Technology and Control, 39*(1), 32-37.

[61] Abdullah, S., & Abdolrazzagh-Nezhad, M. (2014). Fuzzy job-shop scheduling problems: A review. *Information Sciences, 278*(0), 380-407.

[62] Nguyen, S. (2013). *Automatic Design of Dispatching Rules for Job Shop Scheduling with Genetic Programming.* Victoria University of Wellington.

[63] Michalewicz, Z., & Fogel, D. B. (2004). *How to solve it: modern heuristics*: Springer Science & Business Media.

[64] Lin, F.-T. (2002). Fuzzy job-shop scheduling based on ranking level (λ, 1) interval-valued fuzzy numbers. *Fuzzy Systems, IEEE Transactions on, 10*(4), 510-522.

[65] Shahzad, A., Mebarki, N., & IRCCyN, I. (2010). *Discovering dispatching rules for job shop scheduling problem through data mining.* Paper presented at the 8th International Conference of Modeling and Simulation-MOSIM.

[66] Zheng, Y. L., & Li, Y. X. (2012). Artificial bee colony algorithm for fuzzy job shop scheduling. *International Journal of Computer Applications in Technology, 44*(2), 124-129.

[67] Yahyaoui, A., Fnaiech, N., & Fnaiech, F. (2011). A Suitable Initialization Procedure for Speeding a Neural Network Job-Shop Scheduling. *Industrial Electronics, IEEE Transactions on, 58*(3), 1052-1060.

[68] Kaur, D., & Murugappan, M. M. (2008). *Performance enhancement in solving traveling salesman problem using hybrid genetic algorithm.* Paper presented at the Fuzzy Information Processing Society, 2008. NAFIPS 2008. Annual Meeting of the North American.

[69] Victer Paul, P., Moganarangan, N., Kumar, S. S., Raju, R., Vengattaraman, T., & Dhavachelvan, P. (2015). Performance analyses over population seeding techniques of the permutation-coded genetic algorithm: An empirical study based on traveling salesman problems. *Applied Soft Computing, 32*(0), 383-402

[70] Brizuela, C. A., & Sannomiya, N. (1999, 17/08/19999). *A Diversity Study in Genetic Algorithms for Job Shop Scheduling Problems.* Paper presented at the Proceedings of Genetic and Evolutionary Computation Conference (GECCO-99), Morgan Kaufmann, San Francisco, CA.

[71] Bahameish, H. A. (2014). Finding a Cost Effective LNG Annual Delivery Program (ADP) Using Genetic Algorithms.

[72] Dalfard, V. M., & Mohammadi, G. (2012). Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints. *Computers & Mathematics with Applications, 64*(6), 2111-2117.