Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Development of Guided Artificial Bee Colony (GABC) Heuristic for Permutation Flowshop Scheduling Problem (PFSP)

Noor Azizah Sidek[1,*], Salleh Ahmad Bareduan[2], Azli Nawawi[3], Ten Jia Yee[2]

[1] Department of Mechanical Engineering, Centre for Diploma Studies, Universiti Tun Hussein Onn Malaysia, Pagoh Higher Education Hub, Jalan Panchor, 84600 Panchor, Johor, Malaysia
[2] Department of Manufacturing Engineering, Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
[3] Department of Mechanical Engineering Technology, Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia, Pagoh Higher Education Hub, Jalan Panchor, 84600 Panchor, Johor, Malaysia

ARTICLE INFO

Article history:
Received 13 July 2023
Received in revised form 27 October 2023
Accepted 8 November 2023
Available online 22 November 2023

Keywords:
Guided Artificial Bee Colony Algorithm; flowshop scheduling; scheduling optimization

ABSTRACT

The flowshop is the most often used production system in the sector, and several efforts have been made to improve its efficiency. The NEH (Nawaz, Enscore and Ham) heuristics are one of the promising techniques. The range includes using heuristics and metaheuristics. By adopting a modified version of the Artificial Bee Colony (ABC) algorithm, which has the disadvantage of a slow converge speed, this study aims to boost NEH. To find high-quality results with a faster convergence rate, this study developed a strategy to increase the convergence speed of ABC. Because of the significant performance in the makespan value (performance indicator), the Total Greedy was adopted in this study, and the author continued to use it throughout the remainder of the research. This study suggested creating a Guided Artificial Bee Colony (GABC) using the First Job Sequence Arrangement Method and the NEH idea. The investigation was based on Taillard benchmark datasets. According to the findings, ABC frequently gave inconsistent outcomes, but surprisingly, GABC, NEH-based ABC, and ABC consistently produced results that were each 68.75%, 63.33%, and 0.01% better than NEH. Finally, the author can state that this analysis validated ABC's slow convergence problem solutions.

1. Introduction

Making products with the greatest quality possible in the shortest amount of time is the flowshop's main goal. A high volume production system can benefit greatly from this criterion because the flowshop system has a set task flow [1-4]. The flowshop is another objective of a lean system, and it is simpler to manage than other system kinds like Job Shop and Project Shop [5]. The flowshop's repeatability is its most major benefit. It features a linear machine configuration, and the system only performs each work in one direction at each machine [6]. Finding the appropriate work sequence to reduce the makespan is the key problem of using the flowshop [7]. It is essential because

* *Corresponding author.*
E-mail address: noorazizah@uthm.edu.my

https://doi.org/10.37934/araset.33.3.393406

if a manufacturer can produce a product more quickly, it will get on the market earlier and have a better chance of capturing market share [8].

Although NEH is effective, using random selection criteria might increase the likelihood of finding better answers because it is not currently being used [9]. From this vantage point, using metaheuristics to develop a strategy that is more effective than NEH appears to be the next stage.

The notion of NEH is used in certain study to improve the performance of their algorithm because it is particularly effective at solving PFSP. To solve the PFSP problem, Kurdi [3] suggested a memetic algorithm (MA) coupled with a brand-new semi-constructive crossover and mutation operator (MASC). The genetic algorithm (GA), simulated annealing (SA), and NEH algorithm are all combined to form the MASC in the study. Tasgetiren, Eliiyi, Pan, and Kandiller introduced a novel PFH NEH(x) that combines the NEH algorithm and the profile-lifting (PF) constructive heuristic. The primary justification for the team's selection of NEH is that it is a straightforward and efficient heuristic for resolving PSFP. The whale swarm algorithm was presented as a hybrid with DNEH by Wang *et al.,* [8]. The initial solution objective of the makespan is optimized in the study using DNEH.

The Artificial Bee Colony (ABC) method is used in this study because, in comparison to other population-based algorithms, it has less parameters that need to be controlled. The ABC algorithm is also a form of algorithm that is reliable, quick to converge, and versatile [10,11]. Additionally, it has been demonstrated that ABC outperforms Genetic Algorithm, Differential Equations (DE), and Particle Swarm Optimization (PSO) [12-14]. Aside from that, ABC has a solid track record in optimizing permutation flowshop scheduling [15,16].

However, the original ABC approach is known for taking a long time to converge, and this flaw will make it harder to obtain accurate makespan values for PFSP [14,17,18]. The current algorithms must be supplemented with a new mechanism to aid in their convergence to the required performance metrics in order to increase the ABC's effectiveness. Researchers frequently incorporate the idea of the leading heuristics, NEH, in the construction of their algorithm in order to speed up the convergence of the ABC algorithm and based on trends in earlier similar works. Utilizing the NEH arrangement in their proposal and hybridizing the NEH notion in their algorithms is a typical approach. The study's goal is to employ the first job sequence approach to enhance ABC's performance when solving PFSP because the practice of doing so is never documented. The entire procedure is referred to as a novel Guided Artificial Bee Colony (GABC) heuristic designed to solve the PFSP (refer methodology flowchart in Figure 1). The goal of this project is to create a new, enhanced ABC algorithm that includes extra converging mechanisms to boost convergence.

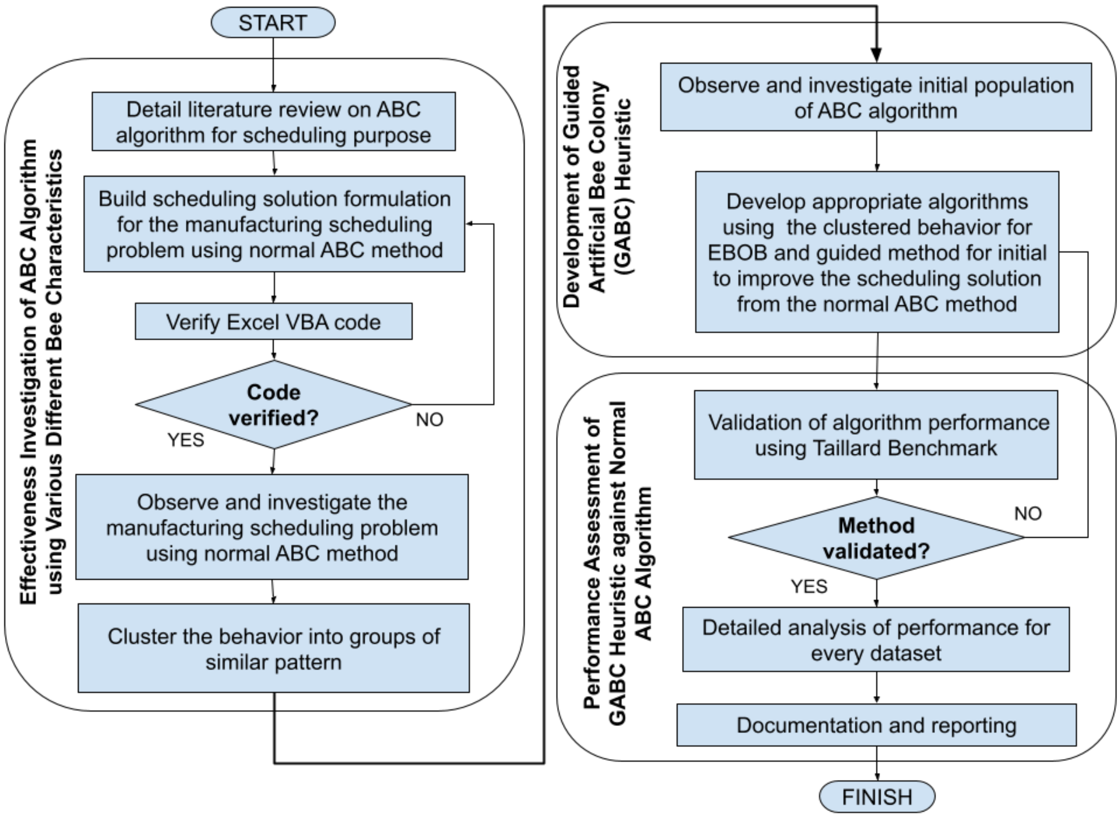

Fig. 1. Methodology to develop Guided Artificial Bee Colony (GABC)

2. Methodology

2.1 Formulation for Permutation Flowshop Scheduling Problem (PFSP)

The makespan, which is seen as an action to reduce the overall completion time for the production operations, is the PFSP optimization aim. The equations for computing the makespan are shown in Eq. (1) to Eq. (4).

Typically, the optimum task permutations to produce the shortest makespan are found using PFSP. There was no stopping when a work started, which was one of PFSP's fundamental rules. The production planner must use the trial-and-error method to determine the order of jobs because the practitioners must also fix the machine sequence (refer Figure 2).

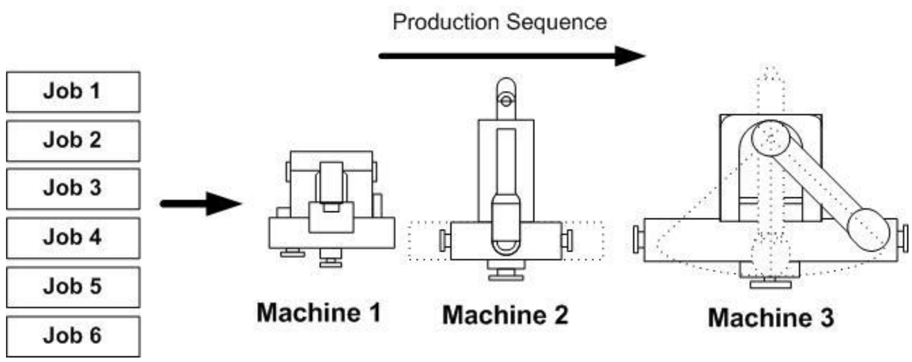

Fig. 2. An illustration of 6 jobs and three machines flowshop [19]

$$C(\pi_1, 1) = p(\pi_1, 1) \tag{1}$$

$$C(\pi_j, 1) = C(\pi_{j-1}, 1) + p(\pi_j, 1) \quad j = 2, \dots, n \tag{2}$$

$$C(\pi_1, k) = C(\pi_1, k-1) + p(\pi_1, k) \quad k = 2, \dots, m \tag{3}$$

$$C(\pi_j, k) = max\{C(\pi_{j-1}, k), C(\pi_j, k-1) + p(\pi_j, k)\} \quad j = 2, \dots, n; \ k = 2, \dots, m \tag{4}$$

Notation:

C: Completion time
π: Job representative
p: Processing time of each job
j: Job identifier
k: Machine identifier
m: Number of jobs
n: Number of machines

PFSP is viewed by researchers as a very challenging optimization issue (NP-hard). Many use heuristics and metaheuristics to tackle the PFSP problem because of its intricacy. The Nawaz-Enscore-Ham (NEH) heuristic is the industry standard for flowshop scheduling. The heuristics, which were created in 1983, make excellent use of the priority order [20]. The NEH heuristic is still widely used as a benchmark today because its dominance and performance that is difficult to match.

2.2 Bee Colony Optimization

By modelling the foraging behavior of bees, the artificial Bee Colony algorithm was first introduced in 2005. The algorithm for the issue of numerical optimization was created by Karaboga [23]. The artificial bee colony approach has been used by researchers to address optimization problems in a variety of fields, including function optimization, software testing, exam scheduling, truss structure optimization, and production schedule optimization.

Three (3) different species of bees cooperate during the foraging phase to locate the optimum food source. The three different kinds of bees are scout bees, onlooker bees, and employed bees (SB). Using this strategy, OB and SB were labelled as unemployed bees.

The employed bee is assigned to one food source by the procedure, and the bee will fly to nearby food sources and memorize which one is the best in the area. When the employed bee gets to the hive, the spectator bees will decide depending on the details provided by the employed bee about the food supply. The information is presented by the bees in the form of a "bee dance," after which the bees will take advantage of the best food source. After multiple iterations, if the food source's fitness (goodness) has not increased, the system will deploy scout bees to look for other food sources. If the new food source's probability value is higher than the current one, the scout bee's discovery will take the place of the latter.

There are three different kinds of bees working together during the optimization process using ABC. The bees are scout bee, employed bee, and spectator bee (SB).

(i) Initialization Phase

Beginning with the initiation stage, ABC develops food sources (solutions). This phase will be carried out by the scout bees (SB), and the system will also set the trial counter's (TC) value at this time. Additionally, this phase also produces solutions as n-dimension vectors., $X_i = \{x_{i,1}, x_{i,2}, \ldots\ldots, x_{i,n}\}$. The solutions are generated using Eq. (5).

$$x_{i,j} = LB_j + r\left(UB_j - LB_j\right) \tag{5}$$

Based on Eq. (5), $x_{i,j}$ is the i^{th} solution of dimension j. The LB_j and UB_j variables stand in for the bottom and upper limits of dimension j, respectively. Additionally, r is a uniform random number between 0 and 1.

(ii) EB Phase

The employed bees (EB) will use their memories to find new solutions (food sources) during the EB Phase. Following that, the EBs will produce potential solutions organized as an n-dimension vector, $Y_i = \{y_{i,1}, y_{i,2}, \ldots\ldots, y_{i,n}\}$. The procedure for producing potential solutions is shown in Eq. (6).

$$y_{i,j} = x_{i,j} + \emptyset\left(x_{i,j} - x_{k,j}\right) \tag{6}$$

In the equation, $x_{i,j}$ is the solutions generated by Eq. (5). $x_{i,j}$and$x_{k,j}$ are in the same dimension, *j* but from a different position. Additionally, Ø is a random number between 0 to 1.

The approach of greedy selection will be used after the solutions have been found. With this approach, the outdated approach is replaced with a fresh, superior one. The waggle dance is used to transmit data about any new food sources, including their position, distance, and fitness value. Bees use the waggle dance to locate food sources without the aid of a map.

(iii) OB Phase

In the OB Phase, observer bees (OBs) will make food source selections using the knowledge gained from the waggle dance. Additionally, the solution with the highest fitness value will be chosen. The probability, P_i for a food source to be selected by the OB is calculated using Eq. (7) [3,17].

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{7}$$

According to the equation, fit_i is the fitness value of food source X_i and *SN* is the number of solutions. Based on the data obtained from the EBs, OBs will produce their version of the solutions. The fitness of the solutions developed by the OBs will be assessed. Similar to this, only the best solution will be chosen using the greedy selection method.

(iv) SB Phase

If a food source cannot be improved after being set by the trial counter multiple times, it will be abandoned (TC). The SB Phase will begin under this scenario. The EB that possesses the abandoned food source will transition to a scout bee (SB) in this phase and proceed in search of a new food source [3,21,22].

ABC will repeat itself till a termination requirement is satisfied. Bee scouts will locate any abandoned food sources.

2.3 ABC Algorithm's Effectiveness Investigated Utilizing Various Bee Characteristics

In contrast to other optimization areas where the system employs the algorithm's formulae to discover the most optimal value, the implementation of ABC in PFSP is significantly different. Only the ABC algorithm's notion is used in this study to change the flowshop's work sequence. The author uses the swap and insert method referred to as ABC to apply the concept of ABC [23]. The conventional way in the PFSP for optimization is to use job reorganisations rather than numerical numbers to express the optimization system.

The author switches the locations of two jobs in order to shift the sequence of the jobs (refer to Figure 3). This will result in an arrangement of jobs in the flowshop, and the author modifies the position of the jobs to insert them (refer to Figure 4).

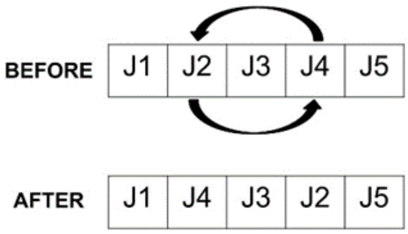

Fig. 3. The system for switching where jobs are located in the flowshop scheduling

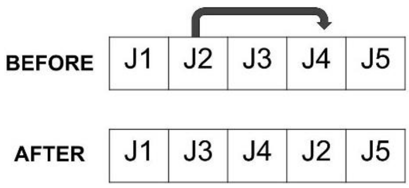

Fig. 4. The mechanism for inserting jobs

2.3.1 Behavior is clustered into a similar pattern

In ABC, the system will abandon using the food source if the outcome does not improve after numerous iterations. The trial counter (TC) determines how many attempts are permitted before giving up the food source. The TC is highly helpful for starting the search process in a new location and is also thought of as the medium to balance the activities of exploration and exploitation.

A high TC value will cause the particles (bees) to hunt in the same region, improving the potential for exploitation. Scout bees are released to search for a new place once the food source has been abandoned; this activity is also referred to as the exploration activity. A lower TC value will promote exploration since scout bees will fly more often in search of new places.

TC was set to three different cycles: six, twelve, and eighteen. 100 data sets were used in the experiment. An error percentage was used to gauge the response (or output). Each TC variation's bee structure is shown in Figure 5. The first set of flow charts represents a scenario where TC is set at six (6) cycles based on the figure. The hive has six bees (3 EBs and 3 OBs). The six bees' best outcomes are contrasted in this scenario. Similar to this, if TC=12 or TC=18 cycles, respectively, the findings will be compared among twelve (12) or eighteen (18) bees. As was already mentioned, improved exploitation activities will be the result of a higher TC value.

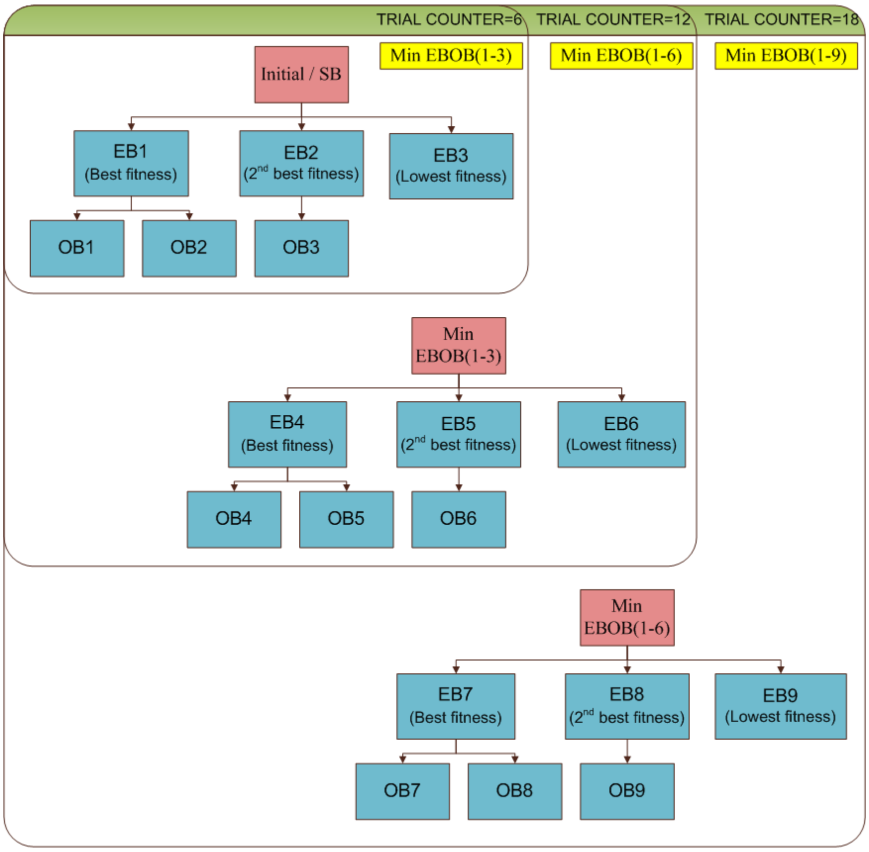

Fig. 5. The ABC approach for PFSP (EB: Employed Bee and OB: Onlooker Bee) [19]

2.4 Development of the GABC Heuristic, a Guided Artificial Bee Colony

The cluster and onlooker behavior were used to build the GABC [24]. The initial solution has been modified in some way. The author continues to make changes to the ABC algorithm to speed up convergence. These changes will direct optimization agents to areas with top-notch solutions.

The procedure will direct the bee population before beginning the optimization to help them perform better. The bees will be able to provide the area with superior solutions due to this. The study employed two methods to accomplish its objective:

(i) NEH-based ABC
(ii) Guided Artificial Bee Colony (GABC)

In a nutshell, NEH-based ABC starts the ABC algorithm with the NEH arrangement solution. The ABC algorithm will be able to investigate superior solutions by starting with NEH. The GABC algorithm starts with the NEH layout solution as an initial step, and the first job sequence approach is added to this step to considerably increase performance.

The study integrated the NEH arrangement as an initial in a population before the optimization utilizing the ABC algorithm in the first choice (NEH-based ABC). In other words, the ABC algorithm will function as usual, but instead of utilizing random initials, it will use the NEH solution because, according to previous research, it is the most effective algorithm for minimizing PFSP. The following is a demonstration of the NEH Algorithm for steps in makespan minimization [3,25,26]:

Step 1: Order the jobs according to the machine's non-increasing processing time totals.
Step 2: Schedule the first two jobs as if they were the only two jobs, and just do your best to reduce the partial makespan.
Step 3: For k=3 to n
Step 4: Place the k^{th} job where it will have the shortest partial makespan among the k available makespans.

The process for guiding the beginning in ABC is depicted in Figure 6. Because NEH is useful for locating the regions with high-quality solutions, optimization efforts were based on the NEH arrangement solution. In NEH-based ABC, the exploration and exploitation processes begin with the NEH results. From this point on, the author anticipated that NEH-based ABC would produce superior outcomes to NEH.

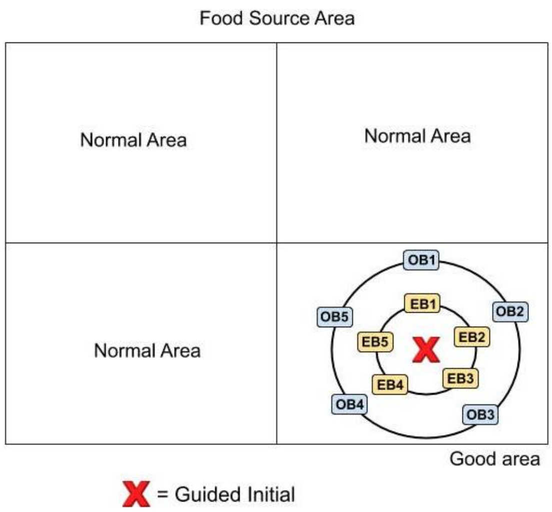

Fig. 6. ABC Guided Initial using the NEH solution (NEH-based ABC)

This study applies the first job sequence approach and the NEH layout solution to the second option (GABC). In the first job sequence approach illustrated in Figure 7, the number in the red boxes is fixed, while the other numbers are chosen at random. The makespan for each column will be determined using the fact that each column represents a job sequence. The goal of this option (GABC) is to help GABC converge more rapidly than the original ABC algorithm by assisting it in locating the areas with high-quality solutions. Figure 8 and Figure 9 exhibit the arrangements depending on the problem, while Table 1 displays an example of the "6 jobs, 3 machines" problem. There are 720 solutions to a problem with 6 jobs associated to it (arrangements).

Arrangement						
1	2	3	4	5	6	
A	B	C	D	E	F	First job sequence
D	F	A	E	D	C	Random
F	C	D	F	C	E	
C	E	B	B	B	A	
B	D	E	A	A	B	
E	A	F	C	F	D	

Fig. 7. Employing a first job sequence approach as an alternative

	Job	Machine		
		1	**2**	**3**
Fixed	**A**	2	4	5
Random	**D**	6	5	1
	F	5	8	6
	C	4	2	9
	B	3	7	6
	E	3	3	4

Fig. 8. Arrangement 1 for the 6 jobs and 3 machines problem

	Job	Machine		
		1	**2**	**3**
Fixed	**B**	3	7	6
Random	**F**	5	8	6
	C	4	2	9
	E	3	3	4
	D	6	5	1
	A	2	4	5

Fig. 9. Arrangement 2 for the 6 jobs and 3 machines problem

Table 1
Example of 6 jobs and 3 machines problem

Job	Machine		
	1	2	3
A	2	4	5
B	3	7	6
C	4	2	9
D	6	5	1
E	3	3	4
F	5	8	6

The pseudocode demonstrates the GABC procedure:

Step 1: Create the first solution using the NEH arrangement
Step 2: Employ the first job sequence approach with the NEH arrangement solution to create the initial population.
Step 3: Assess the solution using the first job sequence population approach.
Step 4: Set cycle to 1
Step 5: Repeat
Step 6: Create new employed bee (EB) solutions utilizing the swap technique and assess the results.
Step 7: Keep the optimum solution to the greedy total (3+0+0) for the observer bee (OB)
Step 8: Using the swap mechanism, create new solutions for the onlooker bee (OB) and assess the results.
Step 9: Use the scout bee (SB) to identify an abandoned food source and replace it (SB).
Step 10: Store the best answer to memory.
Step 11: Cycle = cycle+1
Step 12: Until cycle = MCN

The first solution generated utilizing the NEH arrangement signified the start of the GABC procedure. The initial population is then formed by using the solution with the first sequence approach. The solution is then assessed using the first job sequence population approach. The cycle is set to one, and the previous stages are done once again.

Using the swap mechanism, new solutions are created for the employed bee (EB), and the system then evaluates the solutions. The system maintains the ideal solution for the onlooker bee's (OB) total greed. The swap technique is then used to generate the OB solutions, which the system then assesses. The abandoned food source is then identified from here, and it is restored using the scout bee (SB). The cycle will be continued until the termination requirement is met after the system memorizes the best and most consistent answer. Transferring some random solutions to the following generation for each iteration is the method used in this work to prevent local optima.

3. Computational Results

3.1 Efficiency for 5 Machines and 50 Jobs

The analysis of 50 jobs 5 machines for multiple iterations is the main topic of this section (5000, 2000, 1000 and 500 iterations). A statistical comparison of NEH, NEH-based ABC, ABC, and GABC serves as a summary of this section. The performance for all datasets is also examined.

3.2 Efficiency for 50 Jobs and 5 Machines for All Datasets

All three of the study's parameters are displayed in the main effects plot (Figure 10). Parameters are sometimes referred to as Factors in Design of Experiments (DOE), and the factors in the above chart are Data, Iterations, and Algorithm. Any figure greater than zero for the percentage of improvements is superior to NEH's and vice versa. According to the abovementioned chart, Data 1 performs the best for the data. These findings are the result of 120 runs in Minitab DOE that took data, iterations, and algorithm into considerations.

Due to its position at the top of the chart's iteration section, the iteration chart for the iterations demonstrates that 5000 iterations gave the best performance. The author can extrapolate that the outcomes for iterations are extremely reasonable since the worst iteration is 500.

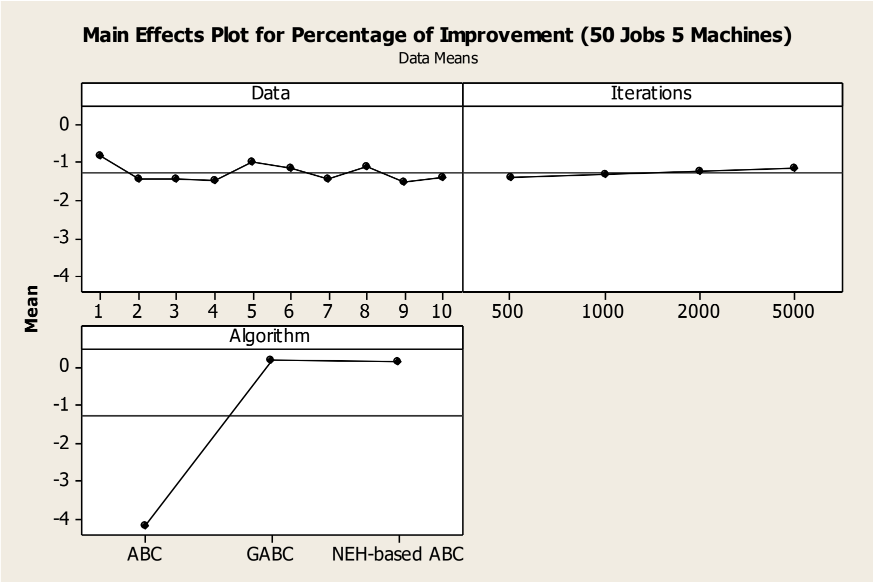

Fig. 10. Main Effects Plot for Improvement Rate (50 jobs and 5 machines)

Both GABC variants, NEH-based ABC and GABC are in the lead positions in the algorithm segment, with GABC performing marginally better. This shows that GABC, the most improved version, is the best, which is a good sign that can lead to a solid conclusion. Additionally, GABC converges faster than its rivals because it required half as many iterations to achieve the best outcomes.

The interval plot showing the percentage of improvement for the NEH-based ABC, GABC, and ABC is shown in Figure 11. The performance of each of the three competitors across the iterations and data set is represented on the chart. The accuracy of each competitor is most readily seen from the chart. The positive value indicates that the generated makespan values are superior to NEH, whilst the negative value indicates that the performance is inferior when compared to NEH. The ABC method produced a high number of negative values, which indicates that nearly all outcomes are inferior to NEH.

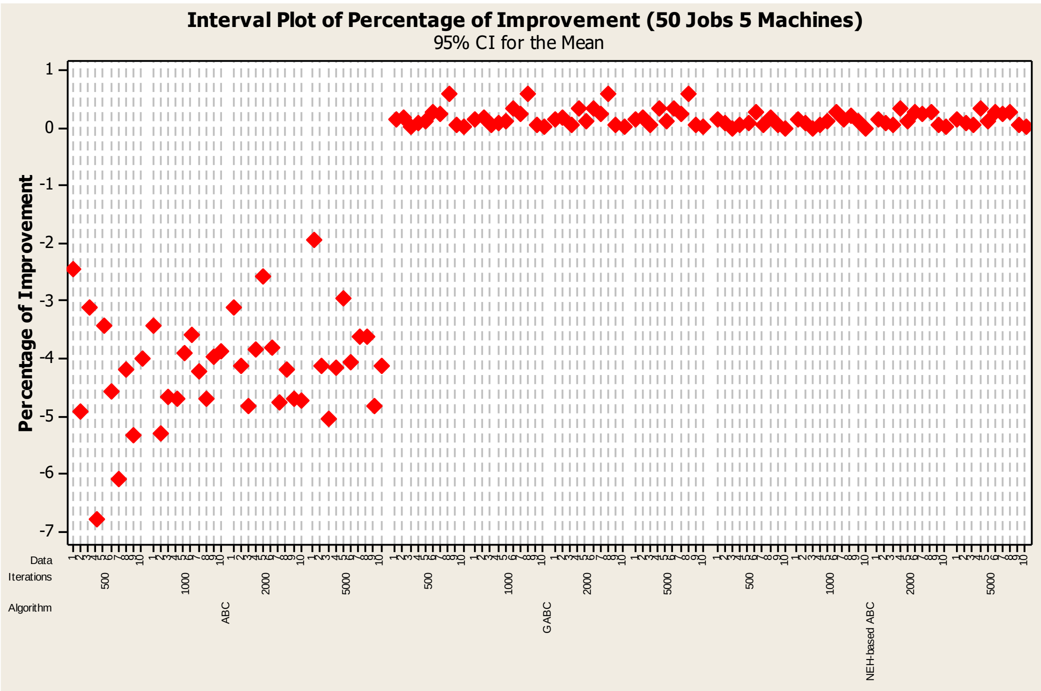

Fig. 11. Interval Plot of Improvement in Percentage (50 jobs and 5 machines)

Furthermore, the precision was reduced due to the results' extreme spread. This is expected since this study tends to make the ABC algorithm's version better. In the meantime, GABC variations that were improved managed to outperform NEH a substantial number of times (points). This indicates that the study's goals are almost fully achieved, which is a highly encouraging indicator.

Additionally, the abovementioned chart demonstrates that GABC has four points with the best makespan. The points are consistent over iterations, demonstrating that GABC outperforms NEH-based ABC in terms of performance. This is due to the fact that GABC is an enhanced version of NEH-based ABC with a new feature like Fixed Initial Solution.

3.3 Statistical Analysis

The statistical comparison of the performances of ABC, NEH-based ABC, and GABC is the main topic of this section. All iterations setting in all datasets are used in the comparison, which is conducted against NEH. Table 2 demonstrates that GABC outperformed NEH across all datasets and iterations. This is due to the fact that GABC combines NEH and ABC with the added benefit of fixed initial positions. With a somewhat worse performance in the lower iteration settings, the NEH-based ABC placed second. This demonstrates that NEH-based ABC converges more slowly than GABC. The slow-to-convergence ABC algorithm was unable to outperform NEH in any iterations setting across all datasets. From here, it can be inferred that NEH performs better than ABC for the scheduling issue of 50 jobs and 5 machines.

Table 2
Comparison of every iteration's percentage to the NEH (50 jobs and 5 machines)

Iterations	Percentage of Comparison to NEH		
	ABC (%)	NEH-based ABC (%)	GABC (%)
5000	0	100	100
2000	0	100	100
1000	0	80	100
500	0	90	100

This study was able to demonstrate that GABC is superior to NEH-based ABC, and the author prepared Table 3 to provide quantitative evidence for this conclusion. The frequency of evidence suggesting GABC is superior to NEH-based ABC is displayed in the table.

Table 3
Justification for GABC better performance than NEH-based ABC

Iterations	Percentage of Data			Range of Percentage of Improvements (Compared to NEH)	
	GABC > NEH-based ABC	GABC = NEH-based ABC	GABC < NEH-based ABC	GABC	NEH-based ABC
5000	50	50	0	0.04 – 0.62	0.04 – 0.36
2000	50	50	0	0.04 – 0.62	0.04 – 0.36
1000	70	20	0	0.04 – 0.62	0.07 – 0.28
500	60	40	0	0.04 – 0.62	0.07 – 0.28

4. Conclusions

This study uses the ABC concept as its primary source of inspiration to solve the PFSP utilising GABC (Swap and Insert mechanism). The community frequently employs this technique when employing metaheuristics to enhance PFSP. The first job sequence arrangement was used by the author to start the optimization process. The NEH arranging solution is implemented by the system to identify the good solutions area, and the ABC algorithm then proceeds from that point. This setting is advantageous because it directs the bees to the superior food source (solution). The EB and OB conduct exploration and exploitation in the area once the original solution identifies it to look for better food sources. In the absence of one, they accept the initial's value.

The study focuses on comparing the performance of NEH, ABC, ABC based on NEH, and GABC. The study utilizes the Taillard Benchmark datasets to make sure the comparison stage is consistent with other related researches. Based on the data, it can be stated that better outcomes will be obtained as the number of iterations increases. This is so that the bees have more time to optimise the answer as the number of iterations increases. It seems to reason that better results will be produced if more time is spent searching (for the most of objective functions). Unfortunately, high iteration counts result in a longer simulation runtime. The majority of decision-makers do not like this trait.

The performance of GABC is evaluated in the validation phase against NEH, ABC, and NEH-based ABC. According to the findings, ABC frequently produced inconsistent results, and the majority of its data are inferior than NEH's. It's interesting to note that while NEH-based ABC's performance decreased after 500 and 1000 iterations, GABC produced 100% outcomes that are the NEH. After conducting this study, some suggestions for subsequent works have been made. Future study is still appropriate for a comprehensive review of EBOB for all issues. Finding the right amount of bees for each scheduling issue is the key goal. It is worthwhile to investigate the possibilities of guiding a few job sequences to obtain superior makespan values in order to obtain the best answers. Create a new approach using reverse engineering and the optimal work arrangements from this study. This may make it possible to obtain outcomes that have a greater quality than those of the present investigation. Finally, the study showed ways to address ABC's slow convergence problem, the author can say.

Acknowledgement
This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through Tier 1 (Vot Q131).

References

[1] Bandyopadhyay, Susmita. *Production and Operations Analysis: Traditional, Latest, and Smart Views*. CRC press, 2019. https://doi.org/10.1201/9781351113670

[2] Fan, Joshua Poh-Onn, and Graham K. Winley. "A Heuristic Search Algorithm for Flow-Shop Scheduling." *Informatica* 32, no. 4 (2008): 453-465.

[3] Kurdi, Mohamed. "A memetic algorithm with novel semi-constructive evolution operators for permutation flowshop scheduling problem." *Applied Soft Computing* 94 (2020): 106458. https://doi.org/10.1016/j.asoc.2020.106458

[4] Pan, Quan-Ke, M. Fatih Tasgetiren, Ponnuthurai N. Suganthan, and Tay Jin Chua. "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem." *Information Sciences* 181, no. 12 (2011): 2455-2468. https://doi.org/10.1016/j.ins.2009.12.025

[5] Park, Sung Hyun. "Scheduling Theory and Problems: Review and Categorization of Solution Procedures." *Journal of the Korean Society of Industrial Engineers* 2, no. 1 (1976): 101-108.

[6] Groover, Mikell P. *Automation, production systems, and computer-integrated manufacturing*. Pearson Education India, 2016.

[7] Gupta, Jatinder N. D., and Edward F. Stafford Jr. "Flowshop scheduling research after five decades." *European Journal of Operational Research* 169, no. 3 (2006): 699-711. https://doi.org/10.1016/j.ejor.2005.02.001

[8] Wang, Guangchen, Liang Gao, Xinyu Li, Peigen Li, and M. Fatih Tasgetiren. "Energy-efficient distributed permutation flow shop scheduling problem using a multi-objective whale swarm algorithm." *Swarm and Evolutionary Computation* 57 (2020): 100716. https://doi.org/10.1016/j.swevo.2020.100716
[9] Emmons, Hamilton, and George Vairaktarakis. *Flow shop scheduling: theoretical results, algorithms, and applications*. Springer Science & Business Media, 2013. https://doi.org/10.1007/978-1-4614-5152-5
[10] Anam, S. "Multimodal optimization by using hybrid of artificial bee colony algorithm and BFGS algorithm." In *Journal of Physics: Conference Series*, vol. 893, no. 1, p. 012068. IOP Publishing, 2017. https://doi.org/10.1088/1742-6596/893/1/012068
[11] Ayan, Kürşat, and Ulaş Kılıç. "Artificial bee colony algorithm solution for optimal reactive power flow." *Applied Soft Computing* 12, no. 5 (2012): 1477-1482. https://doi.org/10.1016/j.asoc.2012.01.006
[12] Karaboga, Dervis, and Bahriye Basturk. "On the performance of artificial bee colony (ABC) algorithm." *Applied Soft Computing* 8, no. 1 (2008): 687-697. https://doi.org/10.1016/j.asoc.2007.05.007
[13] Karaboga, Dervis, and Bahriye Akay. "A comparative study of artificial bee colony algorithm." *Applied Mathematics and Computation* 214, no. 1 (2009): 108-132. https://doi.org/10.1016/j.amc.2009.03.090
[14] Sonmez, Mustafa. "Artificial Bee Colony algorithm for optimization of truss structures." *Applied Soft Computing* 11, no. 2 (2011): 2406-2418. https://doi.org/10.1016/j.asoc.2010.09.003
[15] Luo, Jun, Qian Wang, and Xianghai Xiao. "A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization." *Applied Mathematics and Computation* 219, no. 20 (2013): 10253-10262. https://doi.org/10.1016/j.amc.2013.04.001
[16] Sulaiman, Noorazliza, Junita Mohamad-Saleh, and Abdul Ghani Abro. "A modified artificial bee colony (JA-ABC) optimization algorithm." In *Proceedings of the International Conference on Applied Mathematics and Computational Methods in Engineering*, vol. 1, pp. 74-79. 2013.
[17] Banharnsakun, Anan, Tiranee Achalakul, and Booncharoen Sirinaovakul. "The best-so-far selection in artificial bee colony algorithm." *Applied Soft Computing* 11, no. 2 (2011): 2888-2901. https://doi.org/10.1016/j.asoc.2010.11.025
[18] Liu, Weibo, Yan Jin, and Mark Price. "A new improved NEH heuristic for permutation flowshop scheduling problems." *International Journal of Production Economics* 193 (2017): 21-30. https://doi.org/10.1016/j.ijpe.2017.06.026
[19] Bareduan, Salleh Ahmad, Nur Fazlinda M. Pauzi, Noor Azizah Sidek, and Azli Nawawi. "Manipulating the Onlooker Bee's Behaviour in Artificial Bee Colony Algorithm for Permutation Flowshop Scheduling." *International Journal of Emerging Trends in Engineering Research* 8, no. 1.2 (2020): 180-186. https://doi.org/10.30534/ijeter/2020/2581.22020
[20] Nawaz, Muhammad, E. Emory Enscore Jr, and Inyong Ham. "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem." *Omega* 11, no. 1 (1983): 91-95. https://doi.org/10.1016/0305-0483(83)90088-9
[21] Akay, Bahriye, and Dervis Karaboga. "Artificial bee colony algorithm for large-scale problems and engineering design optimization." *Journal of Intelligent Manufacturing* 23 (2012): 1001-1014. https://doi.org/10.1007/s10845-010-0393-4
[22] Karaboga, Dervis, Celal Ozturk, Nurhan Karaboga, and Beyza Gorkemli. "Artificial bee colony programming for symbolic regression." *Information Sciences* 209 (2012): 1-15. https://doi.org/10.1016/j.ins.2012.05.002
[23] Banharnsakun, Anan, Booncharoen Sirinaovakul, and Tiranee Achalakul. "Job shop scheduling with the best-so-far ABC." *Engineering Applications of Artificial Intelligence* 25, no. 3 (2012): 583-593. https://doi.org/10.1016/j.engappai.2011.08.003
[24] Ruiz, Rubén, Quan-Ke Pan, and Bahman Naderi. "Iterated Greedy methods for the distributed permutation flowshop scheduling problem." *Omega* 83 (2019): 213-222. https://doi.org/10.1016/j.omega.2018.03.004
[25] Kalczynski, Pawel J., and Jerzy Kamburowski. "An improved NEH heuristic to minimize makespan in permutation flow shops." *Computers & Operations Research* 35, no. 9 (2008): 3001-3008. https://doi.org/10.1016/j.cor.2007.01.020
[26] Sidek, N. A., S. A. Bareduan, and A. Nawawi. "Performance investigation of artificial bee colony (ABC) algorithm for permutation flowshop scheduling problem (PFSP)." In *Journal of Physics: Conference Series*, vol. 1150, no. 1, p. 012060. IOP Publishing, 2019. https://doi.org/10.1088/1742-6596/1150/1/012060