Open Access
Published Online First: March, 2023

Baghdad Science Journal
2023, 20(5 Suppl.): 2067-2076

P-ISSN: 2078-8665
E-ISSN: 2411-7986

DOI: https://dx.doi.org/10.21123/bsj.2023.6810

Diversity Operators-based Artificial Fish Swarm Algorithm to Solve Flexible Job Shop Scheduling Problem

Sura Mazin Ali[1]   *Alaa Wagih Abdulqader*[2*]

[1] Political Sciences College, Mustansiriyah University, Baghdad, Iraq
[2] Computer Techniques Engineering Department, Baghdad College of Economic Sciences University, Baghdad, Iraq
*Corresponding author: alaa_wagih@baghdadcollege.edu.iq
E-mail address: Suraaz2007@uomustansiriyah.edu.iq

Received 28/12/2021, Revised 1/11/2022, Accepted 3/11/2022, Published Online First 20/3/2023, Published 28/10/2023

Abstract:

Artificial fish swarm algorithm (AFSA) is one of the critical swarm intelligent algorithms. In this paper, the authors decide to enhance AFSA via diversity operators (AFSA-DO). The diversity operators will be producing more diverse solutions for AFSA to obtain reasonable resolutions. AFSA-DO has been used to solve flexible job shop scheduling problems (FJSSP). However, the FJSSP is a significant problem in the domain of optimization and operation research. Several research papers dealt with methods of solving this issue, including forms of intelligence of the swarms. In this paper, a set of FJSSP target samples are tested employing the improved algorithm to confirm its effectiveness and evaluate its execution. Finally, this paper concludes that the enhanced algorithm via diversity operators has discrepancies about the initial AFSA, and it also provided both sound quality resolution and intersected rate.

Keywords: Combinatorial Optimization Problems, Diversity Operators, Fish Swarm Artificial Algorithm, Flexible Job Shop Scheduling problem, Metaheuristic Algorithm.

Introduction:

Production managing has a great influence on the process of planning and scheduling of contemporary manufacturing systems, scheduling of production involves assigning the source over a period to perform a bunch of tasks, which can be considered one of the most demanding and grave topics in handling and scheming of manufacturing processes[1]. Many studies have been done to overcome several types of scheduling problems, one of the most difficult problems in this field is the job shop scheduling problem (JSSP)[2], which is a highly popular problem in the fields of production management and optimization of combinatorial[3]. It is necessary to have a set of apparatuses to handle many jobs. Every job is made of a preset string of processes in order to perform each process, where one specified apparatus is assigned for one process to handle only one process at a time[4].

In order to improve the classical problem of job scheduling, it has been circulated to an FJSSP, in FJSSP a fixed number. of apparatuses with processing time can operate a job operation not similar in JSSP where each process must be performed by a particular apparatus; this is why when JSSP is compared with FJSSP, it is concluded that the FJSSP is more awkward due to its uncertainty when choosing the right apparatus out of a specific set of apparatuses that would handle each process of a task[5] .Two sub-problems were decomposed from the problem of scheduling jobs in FJSP that are routing and scheduling. A routing sub-problem means each operation is assigned to a particular machine (out of a set of capable machines) and a scheduling sub-problem means that each assigned operation is given a sequence number for all selected machines in order to achieve optimized objectives from the feasible schedule[6,7]. In this research the makespan is decreased i.e., the highest time to complete the tasks. FJSSP is intensely NP-hard [8], and is a further complicated form of JSSP. Accordingly, the metaheuristic algorithms are the vital substitute to resolve this kind of problems which give an appropriate solution within a satisfactory time period [9, 10].

In[11] (2021), a new variable neighborhood descent hybrid genetic algorithm (VND-hGA) was proposed. To increase convergence performance and solution correctness, a barebones particle swarm optimization (BBPSO)-based mutation operator, a hybrid heuristic initialization method, and VND based on an improved multilevel neighborhood structure are introduced into the conventional GA framework. Furthermore, for maximizing the benefits of BBPSO, a real-number-based chromosomal representation, coding, decoding, and crossover method is provided. VND-hGA outperformed current solutions in terms of solution accuracy and convergence. In[12], the squirrel algorithm was used to solve the flexible shop scheduling problem, and the following are the primary innovations: To address working efficiency and system stability, a new dynamic multi-objective flexible shop scheduling model is proposed, with completion time, load balance (LB), and fault rescheduling deviation as objective functions, taking into account the changes in an actual workshop environment, including the random arrival of task sequences, the departure of completed tasks, sudden machine failure, and the repair of failed machines. Secondly an improved dynamic multi-objective squirrel search method based on decomposition is proposed in[12] in order to effectively solve the above stated mathematical model. It uses operation and machine coding for individual coding and optimizes the SSA's individual evolutionary strategy based on the FJSSP's features. In addition, new dynamic processing technology is proposed that includes the establishment and update of the external population as well as the initialization of the population at the time of rescheduling, allowing the system to respond quickly to changes by regenerating the initial population at the time of rescheduling. In[13], presented a novel crossbred metaheuristic method based on Particle Swarm Optimization (PSO) and Tabu Search (TS) processes. It is created to address the issue of flexible job shop scheduling (FJSSP). For the algorithm's experimental inquiry, several benchmark examination cases from various lit origins are utilized. The divergence from optimality yielded a mean error of 1.09 percent. The test findings provide a solid foundation for further PSO-TS enhancement. In[14], an improved AFSA algorithm was proposed for solving the flexible job shop scheduling problem (FJSSP). The enhancement is based on the Variable Neighborhood Descent (VND) method, which is applied to AFSA by various neighborhood structures in order to improve the original AFSA's performance. In addition, an improved algorithm termed (AFSA-VND) was tested for performance evaluation on certain FJSSP benchmark instances. In terms of solution quality and convergence rate, the experimental results suggest that AFSA-VND outperforms the original AFSA. In[15], camel herds algorithm (CHA) was introduced as a new swarm intelligent algorithm. The suggested method is based on camel behavior in the wild, taking into account that each herd has a leader, and that food and water are sought based on a humidity value using a nearby technique. To test the suggested technique, the flexible job shop scheduling problem (FJSSP) is used as a case study. The CHA is a good method for finding the best solution in problem space, according to the testing results. In[16], the cuckoo search algorithm for addressing the flexible job shop scheduling problem (FJSP) has two improvements: the first is based on the best neighbors generation (CS-BNG), and the second is based on Iterative Levy Flight (CS-ILF). To improve searching in discrete state space, some adaptations to the main points of the CS algorithm have been made. The proposed algorithms have improved the quality and rate of convergence of solutions. For performance evaluation, the improved algorithms are evaluated on the HU dataset for FJSP benchmark instances. In comparison to the basic cuckoo search algorithm, the experimental findings show that the upgraded algorithms are more effective.

This paper tried to enhance the artificial fish swarm algorithm (AFSA) using a diversity operator to solve the (FJSP) complications. The rest of the paper is organized as follows: Section 2 reviews some associated research. Section 3 provides a broad description of AFSA. The improved AFSA is suggested in Section 4, and the results obtained experimentally, and their discussion are presented in Section 5. Eventually, the conclusions reached in this paper are presented in Section 6.

Materials and Methods:

Artificial Fish Swarm Algorithm

Lei introduced AFSA in 2002 as a swarm algorithm that relies on metaheuristics. The general idea of AFSA is to replicate the swarming of fish in the environment; one of the techniques which are used to solve various problems for ideal solutions is to adapt the intelligence of a fish swarm with artificial intelligence[17, 18]. Naturally in a marine habitat, a single fish or a group of fish will find areas that are rich in more nutrition. In the behavior of fish-eating, the behavior of the swarm has a significant effect in facilitating and simplifying the search for food. Because some aspects of nutrition are first discovered by some individuals, who are usually the first to use this nutrition, this simplifying effect arises. When the eating fish appear, other individuals get attracted, and within a short period, the entire swarm quickly huddles in that area of food and starts excessive eating[18]. In a line of this performance, the Artificial Fish (AF) [19-23] replica expresses a set of fish performances such as the following, teeming, and predatory performance for probing in the problem space. The AF model is made up of two parts: variables and functions. The variables section contains the following items: X denotes the current position of AF, which is represented as an array $X = [x1, x2 \dots xn]$, step denotes the maximum length of the movement step, and the functions component covers three AF behaviors: preying behavior, swarming behavior, and following behavior[24-27]. Visual refers to the visual distance of AF, try number is the maximum number of searching tours within the visible distance, and a crowd factor (δ) ($0 < \delta < 1$). The basic steps of the AFSA algorithm are depicted in algorithm 1.

Algorithm 1. AFSA

Begin
For each AF_i *(Artificial Fish) Do*
  Initialize X_i $(i = 1, 2, \dots, n)$;
  Compute fitness function $f(X_i)$;
  End For
Leaflet $= Min\,(f(X_i))$;
Repeat
For each AFi Do
  Carry out Prey behavior on X_i;
  Compute $X_{i,prey}$;
  Carry out Swarm behavior on X_i;
  Compute $X_{i,swarm}$;
  Carry out Follow behavior on X_i;
    Compute $X_{i,follow}$;
$X_i = Best\ of\ (X_{i,prey}, X_{i,swarm}, X_{i,follow})$;
    End For
  $X_{i,best} =$ *Best Solution from* X_i;
  If $f(X_{best}) < f(Leaflet)$ *then*
  Leaflet $= X_{best}$
  Until *(Stopping Criteria are met)*;
  Best Solution = Leaflet;
End.

A. Preying behavior

It is a vital life living conduct of fish for nutrition inclination; fish typically use their vision to sense the condensation of food in water and determine the direction of movement that they can select accordingly. To describe the above behavior: Suppose that X_i is the current state of AF, X_j^t is the AF status random in a visual distance, (X) refers to food condensation (AF value is the objective function). The following state of $AF\ (X_j)$ is calculated as

$$X_j = X_i + visual \times rand\,() \qquad 1$$

Where, random ( ) function is a built-in function to generate random numbers between 0 and 1. If the result of equation 1 is better than the existing value $AF\ (X_i)$, so moving forward direction as in equation 2.

$$X_i^{(i+t)} = X_i^t + \frac{X_j - X_i^t}{||X_j - X_i^t||} \times \text{step} \times \text{rand}\,() \qquad 2$$

Where, step is the length of step.

Then, randomly selecting an alternative X_j status and testing if it has a priority in the value of the objective function to meet the progress requirements. If it is not able to obtain an improved result that meets the progress requirements after a number of trial times, a random step will be taken

B. Swarming behavior

The transition of fish in nature takes place in the form of aggregates to avert risks and maintain the sustainability of the colony. The following diagram depicts the swarming behavior: Assume that the current state of AF is X_i and that X_c is the center place, where the number of its neighbors in the visual field is nf, where ($\|X_j - X_i\| < visual$), n is the overall number of fish, and F(X_c) is better than $F(X_i)$, and ($nf\,/\,n < \delta$), refers to extra food in the neighbors center (a higher value of fitness function in the center and not too congested), so it moves a step

$$X_i^{(t+1)} = X_i^t + \frac{X_c - X_i^t}{||X_c - X_i^t||} \times \text{step} \times rand\,() \qquad 3$$

If not, the preying behavior is performed.

C. Following behavior

During the moving of fish swarm stage, whether a single fish or a group of fish found the food, their neighbors will move towards and get nutrition hastily. This behavior can be described as: X_i is the current state of AF, and X_j is the neighborhood, where ($\|X_j - X_i\| < visual$), means it has better value for fitness function $F(X_j)$. If $F(X_j)$ is better than $F(X_i) \times (nf\,/\,n < \delta)$ this implies that there is a greater food condensation on the neighbor's place and that the surrounding is not very congested, so it proceeds a step toward the neighbor X_j.

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_j - X_i^{(t)}}{||X_j - X_i^{(t)}||} \times \text{step} \times \text{rand}\,() \qquad 4$$

If not, preying behavior is executed.

AFSA Based on Diversity Operators:

Representation of Solutions

Operation sequence and machine assignment are two sub-problems of FJSSP. The solution must be described in matrices such as V_1 and V_2; where V_1 denotes the operation sequence while V_2 denotes the machine assignment. An example for the FJSSP with three jobs and three machines is shown in Fig. 1. The operation sequence can be represented by the permutation approach. Gen et al. produced an approach for operation sequence solution vector representation using feasible permutation [14]. The same symbol is used to name all operations of a job, that symbol is the job number. Depending on the order in which the job shows in the (V_1). The operation of a job number is selected i.e., number of operations that this job content is equal to the job number shown in (V_1). This representation always has forms of convenient operation sequence because there is no break for the priority constraints.

Sequence Vector (V_1)

Jobs	1	3	3	2	1	2
Operations	1	1	2	1	2	2

Assignment Vector (V_2)

Machines	3	1	2	3	2	2

Figure 1. Solution Representation of FJSSP.

A Modified AFSA Based on Diversity Operators

In an attempt to enhance the basic AFSA performance, an AFSA based on diversity operators (AFSA-DO) is proposed in this paper. AFSA is a good algorithm in the optimization problem, but sometimes there are several solutions that need more diversification to increase the performance and decrease the execution time. The diversity operators are a collection of operators aims to produce updated neighbor solutions for several purposes such as:

- Various solutions for the purpose of exploring the largest number of them.
- Reducing the effect of the local minima problem.
- Exploitation is a diversification to increase the solutions space, subsequently, increases the good solution(s).
- Reducing the randomization of AFSA to exploit the nearest.

The proposed AFSA-DO main steps are shown meanwhile in the below algorithm. The maximum number of solution locations that switch their position in the solution vectors is depicted in the visual idea of the AFSA-DO. Hamming distance is used to compute the distance between (Xi, Xj) solutions where the corresponding positions have dissimilar values. In the discrete space, the AF movement step forward will take the same solution that is estimated as the next better solution for each AF. The basic steps of the Improved AFSA via Diversity Operators are illustrated in algorithm 2.

Algorithm 2. Improved AFSA via Diversity Operators

$\boldsymbol{Begin}$
$For\ each\ AF_i\ (Artificial\ Fish)\ Do$
$Initialize\ X_i\ (i = 1, 2, \dots, n);$
$Compute\ fitness\ function\ f(X_i);$
$\boldsymbol{End\ For}$
$Leaflet\ =\ Min\ f(X_i)\ ;$
$\boldsymbol{Repeat}$
$For\ each\ AF_i\ Do$
$Carry\ out\ Prey\ behavior\ on\ X_i;$
$Compute\ X_{i,prey};$
$Carry\ out\ Swarm\ behavior\ on\ X_i;$
$Compute\ \ X_{i,swarm};$
$Carry\ out\ Follow\ behavior\ on\ X_i;$
$Compute\ X_{i,follow};$
$X_i\ =\ Best\ of\ (X_{i,prey}\ , X_{i,swarm}\ , X_{i,follow});$
$\boldsymbol{End\ For}$
$X_{i,best}\ =\ Best\ Solution\ from\ X_i;$
$Carry\ out\ Diversity\ Operators\ on\ \ X_{i,best}\ to\ produc$
$Leaflet\ =\ Best\ of\ (X_{best}, X_{i,DO});$
$\boldsymbol{Until}\ (Stopping\ Criteria\ are\ met);$
$Best\ Solution\ =\ Leaflet;$
$\boldsymbol{End}.$

In AFSA-DO, diversity operators are suggested to implement on the best AF that has been detected till now in each generation (repetition). A diversity operators' strategy is offered to take a better AF solution as input to further condense around the best resolution area in the state space. Five neighborhood operator structures are used in the AFSA-DO strategy which are:

- A mutation operator: generate neighbors based on a small mutation operator.
- An insertion operator: generate neighbors based on the insertion small piece of the solution instead of other pieces.
- A translocation operator: generate neighbors by replacing a small piece of a solution with another piece.
- An inversion operator: generate neighbors based on an inverse small piece of the solution.
- A swapping operator: generate neighbors based on swapping some positions of the solution by other positions.

The above 5 neighborhood strategies depend on the position-based neighborhood (mutation, insertion, and translocation operator) and order-based neighborhood (inversion and swapping operators) for better solution(s) finding. The buffer records DO solution if the solution that has been obtained from diversity operator strategies is better than that from the buffer. Using DO operators with a diversity of neighborhood structures for searching allows the proposed AFSA-DO algorithm to exploit more of the area around the global best AF, resulting in a better solution and faster convergence. The algorithm below illustrates the essential steps used by diversity operators to improve AFSA.

Algorithm 3. Diversity Operators Algorithm

$\boldsymbol{Begin}$
$\boldsymbol{Repeat}$
$\#Choice\ one\ from\ the\ following\ diversity\ operato$
$Case$ 1:
$Mutation\ operator\ applies\ on\ X_{\text{best}}$
$to\ obtain\ local\ optimum\ X_{\text{optimum}}\ mutate-neighbors\ of\ X_{best};$
$Case$ 2:
$Insertion\ operator\ applies\ on\ X_{\text{best}}$
$to\ obtain\ local\ optimum\ X_{\text{optimum}}\ insert-neighbors\ of\ X_{best};$
$Case$ 3:
$Translocation\ operator\ applies\ on\ X_{\text{best}}$
$to\ obtain\ local\ optimum\ X_{\text{optimum}}\ translocate-neighbors\ of\ X_{best};$
$Case$ 4:
$Inversion\ operator\ applies\ on\ X_{\text{best}}$
$to\ obtain\ local\ optimum\ X_{\text{optimum}}\ inverse-neighbors\ of\ X_{best};$
$Case$ 5:
$The\ swapping\ operator\ applies\ on\ X_{\text{best}}$
$to\ obtain\ a\ local\ optimum\ X_{\text{optimum}}\ swap-neighbors\ of\ X_{best};$
$\boldsymbol{End\ Case}$
$\boldsymbol{Until}\ (f(X_{\text{best}}) < f(X_{\text{optimum}}))$
$X_{\text{best}} = X_{\text{optimum}};$
$\boldsymbol{Return}\ (X_{best});$
$\boldsymbol{End}.$

Experimental Results

In order to measure the performance and power of the proposed algorithm (AFSA-DO), several experiments have been executed using the FJSSP dataset and comparing the performance between the AFSA-VND and the original AFSA[25]. With the original AFSA, the artificial fish are moved from one location to another in the continuous state space. The AFSA and AFSA-DO algorithms are subjected to some adaptation to be appropriate for diversification in the discrete state space of the problem using many diversification operators. In this work, the MATLAB platform is employed to execute AFSA and AFSA-DO with computer specifications: Intel Core i7, 2.7 GHz, 4 GB RAM, 500 GB hard drive and Windows 7 have used these algorithms[25], which are made up of three separate instance sets: "edata," "rdata," and "vdata," with the customizable machines set being increased by a probability distribution in each case. Size (j) jobs and (k) machines are assigned for each instance, where $(j \times k)$ will equal to sequence vector and assignment vector. The same FJSSP instances with the same parameter value are used in these three algorithms (AFSA, AFSA-DO, and AFSA-VND) [14] and the parameters are initialized as follows:

$$If\ (j \times k) \le 33, population\ size = 12, Visual = 8.$$
$$If\ (33 < (j \times k) \le 51, population\ size = 14, Visual = 9.$$
$$If\ (51 < (j \times k) \le 80, population\ size = 17, Visual = 12.$$
$$If\ ((j \times k) > 80, population\ size = 21, Visual = 14.$$
$$Trial\_No = 2 \times Visual.$$
$$Maximum\_Gen = 1200$$

Table 1 shows the results of the test runs, with the best-known Lower Bound of dataset (each occurrence) represented by (LB) [26], the mean of makespan for each occurrence represented by Average makespan (AMK) which is obtained from the test runs for the three algorithms mentioned above, and the relative error percentage represented by (REP) for the three algorithms under study, which computed as:

$$REP = [(AMK - LB) / LB] * 100 \qquad 5$$

Table 1. Optimal Mean of Original AFSA and Some Modifications

Instances	LB	AFSA		AFSA-DO		AFSA-VND	
		AMK	REP	AMK	REP	AMK	REP
edata_mt06	55	55	0	55	0	55	0
edata_mt10	871	985	13.08	914	4.93	953	9.41
edata_la1	609	620	1.80	612	0.49	617	1.31
edata_la2	655	691	5.49	663	1.22	686	4.73
edata_la3	550	578	5.09	559	1.63	568	3.27
edata_la4	568	606	6.69	573	0.88	598	5.28
edata_la5	503	518	2.98	511	1.59	511	1.59
edata_la6	855	860	3.24	850	1.4	850	2.04
edata_la7	762	811	6.43	779	2.23	799	4.85
edata_la8	845	869	2.84	854	1.06	860	1.77
rdata_mt06	47	48	2.12	47	0	47	0
rdata_mt10	679	848	24.8	793	16.79	824	21.35
rdata_la1	570	605	6.14	581	1.93	590	3.50
rdata_la2	529	563	6.42	547	3.4	554	4.72
rdata_la3	477	509	6.70	488	2.3	497	4.19
rdata_la4	502	540	7.56	513	2.19	528	5.17
rdata_la5	457	482	5.47	461	0.87	476	4.15
rdata_la6	799	827	3.50	809	1.25	816	2.12
rdata_la7	749	787	5.07	754	0.66	767	2.40
rdata_la8	765	797	4.18	772	0.91	788	3.00
vdata_mt06	47	47	0	47	0	47	0
vdata_mt10	655	773	18.01	697	6.41	746	13.89
vdata_la1	570	599	5.08	581	1.93	590	3.50
vdata_la2	529	579	9.45	548	3.59	562	6.23
vdata_la3	477	508	6.49	497	4.19	503	5.45
vdata_la4	502	536	6.77	520	3.58	527	4.98
vdata_la5	457	484	5.90	469	2.62	479	4.81
vdata_la6	799	829	3.75	811	1.5	819	2.50
vdata_la7	749	781	4.27	767	2.4	773	3.20
vdata_la8	765	785	2.61	773	1.04	780	1.96
AREP			**6.07**		**2.435**		**4.38**

The above outcomes indicate that for all instances applied in the experiment, the makespan obtained by AFSA-DO is superior to that of the original algorithm of AFS and modified AFSA based on VND, and it has been approximated above the lower bound. Moreover, AREP is greater than (AREP) of AFSA-DO for both AFS and AFSA based on VND as original algorithms.

Furthermore, when comparing the number of iterations for finding makespan using AFSA-DO versus AFSA, it is found that the number of iterations of the instance applied into the test is 82% higher than the number of iterations of the instance of AFSA-DO, and in some cases, the makespan is generated by AFSA-DO with significantly fewer iterations than AFSA.

Table. 2, shows a comparison between the results obtained through the suggested algorithm (AFSA-DO) with those attained by the algorithms suggested in [14, 16]. By applying equation 5 on the dataset[21] to calculate the REP of the makespan for the proposed algorithm and some others[21]. To illustrate that, Table. 2, clarifies that REP of the modification of AFSA based on Diversity Operators is not worse than REP; of most instances for the algorithms executed in (AFSA-VND, CS-BNG and CS-ILF) [14, 16]; in addition, the results showed that AREP in the proposed algorithm (AFSA-DO) is lower than the other algorithms that are compared. Figs. 2, 3, illustrate the comparison between the got effects.

Table 2. Mean REP for algorithm AFSA-DO and a few others

Instances	REP			
	AFSA –DO	AFSA-VND	CS -BNG	CS -ILF
edata_mt06	0	0	0	0
edata_mt10	4.93	9.41	13.2	12.39
edata_la1	0.49	1.31	4.43	4.10
edata_la2	1.22	4.73	7.93	5.95
edata_la3	1.63	3.27	7.81	6.90
edata_la4	0.88	5.28	9.15	8.97
edata_la5	1.59	1.59	4.37	4.57
edata_la6	1.4	2.04	3.72	3.36
edata_la7	2.23	4.85	7.34	7.48
edata_la8	1.06	1.77	4.14	2.72
rdata_mt06	0	0	6.38	6.38
rdata_mt10	16.79	21.35	18.11	17.96
rdata_la1	1.93	3.50	6.49	6.84
rdata_la2	3.4	4.72	8.31	7.18
rdata_la3	2.3	4.19	8.59	7.33
rdata_la4	2.19	5.17	7.96	7.17
rdata_la5	0.87	4.15	5.90	5.03
rdata_la6	1.25	2.12	4.13	2.75
rdata_la7	0.66	2.40	4.00	3.60
rdata_la8	0.91	3.00	3.66	3.26
vdata_mt06	0	0	4.25	2.12
vdata_mt10	6.41	13.89	13.89	11.29
vdata_la1	1.93	3.50	7.54	6.84
vdata_la2	3.59	6.23	6.80	6.61
vdata_la3	4.19	5.45	7.96	9.01
vdata_la4	3.58	4.98	6.37	5.77
vdata_la5	2.62	4.81	6.12	9.19
vdata_la6	1.5	2.50	3.37	2.75
vdata_la7	2.4	3.20	3.33	3.20
vdata_la8	1.04	1.96	1.83	2.87
AREP	**2.435**	**4.38**	**6.57**	**6.12**

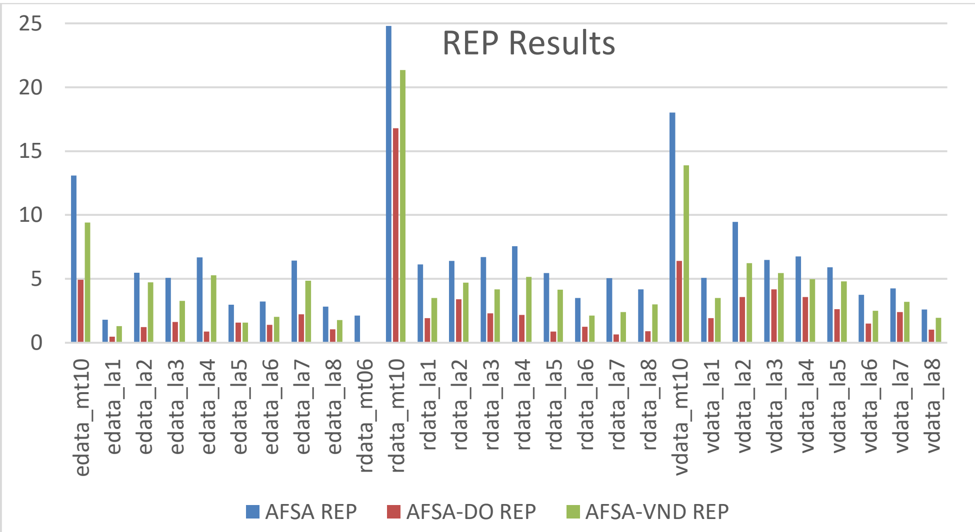

Figure 2. REP results of AFSA, AFSA-DO, and AFSA-VND.

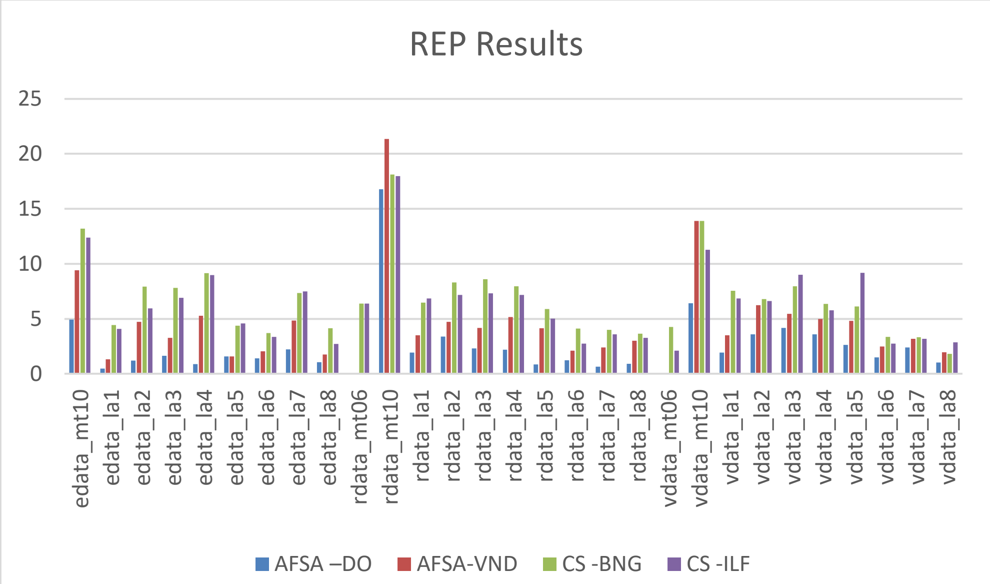

Figure 3. REP results of AFSA, AFSA-DO, AFSA-VND, CS-BNG, and CS-ILF.

Conclusion and Future Work

This paper presents a modification on the AFSA based on diversity operators. Moreover, the diversity operators (mutation, insertion, translocation, inversion, and swapping) provide more diversification, enforcement, and utilization than the basic AFSA. Also, using different operators can increase diversification for current solutions and utilization of the local optimum, which maximize the convergence ratio. The AFSA local search ability has been improved through this proposed algorithm. The laboratory effects made it clear that the improvement made to the proposed AFSA-DO led to more excellent quality solutions for all the instances that underwent the test. Moreover, in most instances, this improvement accelerated the convergence ratio in most cases compared to the original AFSA and AFSA-VND. This affords AFSA-DO the prevalence over the original AFSA and AFSA-VND. In the future, this work will be designed utilizing other algorithms and techniques, and a comparison of effects will be made.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Baghdad College of Economic Sciences University.

Authors' contributions statement:

S.M A. Conception, design, acquisition of data, analysis. A.W. A. Acquisition of data, Interpretation of data, drafting and revising of the article.

References:

1. Zhu Z, Zhou X. An efficient evolutionary grey wolf optimizer for multi-objective flexible job shop scheduling problem with hierarchical job precedence constraints. Comput Ind Eng. 2020; 140: 106280. https://doi.org/10.1016/j.cie.2020.106280
2. Gao D, Wang G, Pedrycz W. Solving Fuzzy Job-Shop Scheduling Problem Using DE Algorithm Improved by a Selection Mechanism. IEEE Trans Fuzzy Syst. 2020; 28(12):3265 - 3275. https://doi.org/10.1109/TFUZZ.2020.3003506
3. Bharti P, Jain S. Hybrid frameworks for flexible job shop scheduling. Int J Adv Manuf Technol. 2020; 108(5-6): 1563–1585. https://doi.org/10.1007/s00170-020-05398-4
4. Abu-Srhahn A, Al-Hasan M. Hybrid Algorithm using Genetic Algorithm and Cuckoo Search Algorithm for Job Shop Scheduling Problem. Int J Comput Sci. 2015; 12(2): 288-292.
5. Salem I E, Mijwil M M, Abdulqader A W, Ismaeel M M. Flight-Schedule using Dijkstra's Algorithm with Comparison of Routes Finding. Int J Electr Comput. 2022; 12(2): 1675-1682. http://doi.org/10.11591/ijece.v12i2.pp%25p.
6. Dehghan-Sanej K, Eghbali-Zarch M, Tavakkoli-Moghaddam R, Sajadi SM, Sadjadi SJ. Solving a new robust reverse job shop scheduling problem by meta-heuristic algorithms. Eng Appl Artif Intell. 2021 May; 101: 104207. https://doi.org/10.1016/j.engappai.2021.104207.
7. HNK Al-behadili. Improved Firefly Algorithm with Variable Neighborhood Search for Data Clustering. Baghdad Sci J. 2022 April; 19(2): 409-421. https://dx.doi.org/10.21123/bsj.2022.19.2.0409.
8. Liu Z, Wang J, Zhang C, Chu H, Ding G, Zhang L. A

hybrid genetic-particle swarm algorithm based on multilevel neighbourhood structure for flexible job shop scheduling problem. Comput Oper Res 2021; 135: 105431. https://doi.org/10.1016/j.cor.2021.105431

9. Osaba E, Villar-Rodriguez E, Ser J D, Nebro A J, Molina D, LaTorre A, et al. A Tutorial On the design, experimentation and application of metaheuristic algorithms to real-World optimization problems. Swarm Evol Comput. 2021; 64: 100888. https://doi.org/10.1016/j.swevo.2021.100888
10. Yasear, S A, Ku-Mahamud K R. Taxonomy of Memory Usage in Swarm Intelligence-Based Metaheuristics. Baghdad Sci J. 2019; 16: 445–452. https://doi.org/10.21123/bsj.2019.16.2(SI).0445.
11. Mijwil M M, Abttan R A. Applying Genetic Algorithm to Optimization Second-Order Bandpass MGMFB Filter. Pertanika J Sci Technol. 2020; 28 (4): 1413–1425. https://doi.org/10.47836/pjst.28.4.15
12. Wang Y, Han J. A FJSSP Method Based on Dynamic Multi-Objective Squirrel Search Algorithm. Int J Antennas Propag. 2021; Article ID 6062689: 1-19. https://doi.org/10.1155/2021/6062689
13. Toshev L. A. Hybrid PSO and TS Algorithm for FJSSP. 10th Natl. Conf. Int. Particip. ELECTRON. 2019 - Proc. 1-6, Sofia, Bulgaria. https://doi.org/10.1109/Electronica.2019.8825612
14. Alobaidi A T S, Hussein S A. An improved Artificial Fish Swarm Algorithm to solve flexible job shop. Ann Conf on New Trends in Info Comm Tech Appli. 2017; 1-6, Baghdad, Iraq. https://doi.org/10.1109/NTICT.2017.7976155
15. Al-Obaidi A T S, Abdullah H S, Ahmed Z O. Camel Herds Algorithm: A New Swarm Intelligent Algorithm to Solve Optimization Problems. I J P C C. 2017; 3(1): 6-10. https://doi.org/10.31436/ijpcc.v3i1.44
16. Al-Obaidi A T S, Hussein S A. Two Improved Cuckoo Search Algorithm to Solve Flexible Job Shop Scheduling Problem. I J P C C. 2016; 2(2): 25-31.
17. Shaker A S, Abdulqader A W, Mijwil M M. DE-striping hype spectral Remote Sensing Images using Deep Convolutional Neural Network. Asian J Appl Sci 2021; 9 (4): 285-290. https://doi.org/10.24203/ajas.v9i4.6719
18. Pythaloka D, Wibowo A T, Sulistiyo M D. Artificial fish swarm algorithm for job shop scheduling problem. In Proc on Int Conf Inf Commun Technol. 2015; 1-6, Nusa Dua, Bali, Indonesia, https://doi.org/10.1109/ICoICT.2015.7231465
19. Feng Y, Zhao S, Liu H, Analysis of Network Coverage Optimization Based on Feedback K-Means Clustering and Artificial Fish Swarm Algorithm. IEEE Acc 2020; 42864 - 42876. https://doi.org/10.1109/ACCESS.2020.2970208
20. Xu H, Zhao Y, Ye C, Lin F. Integrated optimization for mechanical elastic wheel and suspension based on an improved artificial fish swarm algorithm. Adv Eng softw. 2019; 137: 102722. https://doi.org/10.1016/j.advengsoft.2019.102722
21. Li X, Keegan B, Mtenzi F. Energy Efficient Hybrid Routing Protocol Based on the Artificial Fish Swarm Algorithm and Ant Colony Optimisation for WSNs. Sens. 2018; 18(10): 3351, https://doi.org/10.3390/s18103351
22. Ma C, He R. Green wave traffic control system optimization based on adaptive genetic-artificial fish swarm algorithm. Neural Comput Appl. 2019; 31: 2073–2083.https://doi.org/10.1007/s00521-015-1931-y
23. Zheng Z, Li J, Duan P. Optimal chiller loading by improved artificial fish swarm algorithm for energy saving. Math Comput Simul. 2019; 155: 227-243. https://doi.org/10.1016/j.matcom.2018.04.013
24. Jia B, Hao L, Zhang C, Huang B. A Privacy-sensitive Service Selection Method Based on Artificial Fish Swarm Algorithm in the Internet of Things. Mob Netw Appl 2020; 26: 1523–1531. https://doi.org/10.1007/s11036-019-01488-0
25. Zong X, Wang C, Du J, Jiang Y. Tree hierarchical directed evacuation network model based on artificial fish swarm algorithm. Int J Mod Phys C 2019; 30(11): 195C097. https://doi.org/10.1142/S0129183119500979
26. Tan W, Mohamad-Saleh J. Normative fish swarm algorithm (NFSA) for optimization. Soft Comput. 2019; 24: 2083–2099. https://doi.org/10.1007/s00500-019-04040-0
27. Hurink J, Jurisch B, Thole M. Tabu search for the job-shop scheduling problem with multi-purpose machines. OR Spectr. 1994; 15: 205-215.

خوارزمية سرب الأسماك الاصطناعية المعتمدة على عوامل التنوع لحل مشكلة جدولة ورشة العمل المرنة

سرى مازن علي[1]   الاء وجيه عبد القادر[2]

[1] كلية العلوم السياسية، الجامعة المستنصرية، بغداد، العراق
[2] قسم هندسة تقنيات الحاسوب، كلية بغداد للعلوم الإقتصادية الجامعة، بغداد، العراق

الخلاصة :

تعد خوارزمية سرب الأسماك الاصطناعية (AFSA) واحدة من خوارزميات السرب الذكية الحاسمة. لذا عمل المؤلفون في هذا البحث على تعزيز AFSA عبر مشغلي التنوع (AFSA-DO). سيقوم مشغلو التنوع بإنتاج حلول أكثر تنوعًا لـ AFSA للحصول على قرارات معقولة. تم استخدام AFSA-DO لحل مشاكل جدولة محل العمل المرنة (FJSSP). ومع ذلك ، فإن برنامج FJSSP يمثل مشكلة كبيرة في مجال التحسين وبحوث العمليات حيث تناولت العديد من المقالات البحثية طرق حل هذه المشكلة ، بما في ذلك أشكال ذكاء الأسراب. في هذا البحث ، تم اختبار مجموعة من عينات الهدف FJSSP باستخدام الخوارزمية المحسنة لتأكيد فعاليتها وتقييم تنفيذها. وتم الاستنتاج بأن الخوارزمية المحسّنة عبر مشغلي التنوع بها فروقات عن AFSA الأولية ، كما أنها قدمت أيضًا دقة جودة سليمة ومعدل تقاطع سليم.

الكلمات المفتاحية: مشاكل التحسين التجميعي، عوامل التنوع، الخوارزمية الاصطناعية لأسراب الأسماك، مشكلة جدولة ورشة العمل المرنة، الخوارزمية المتعصبة.