

Application project task scheduling using dolphin swarm technology

Baraa S. Mostafa, Firdews A. Alsalman

Department of software Engineering, College of Computer Science and Mathematics, Universiti of Mosul (UOM), Iraq

Article Info

Article history:

Received Dec 3, 2020

Revised Jun 8, 2021

Accepted Jun 15, 2021

Keywords:

Dolphin intelligent algorithm

Project scheduling

Software engineering

Swarm intelligence algorithms

ABSTRACT

Project task scheduling, which is part of software project management, has great importance in the success or failure of projects as 40% of projects are subject to failure due to poor planning and misuse of human resources, and this greatly confirms that the wise decisions that are made at this stage lead to reduced Obstacles facing work and leading to success. The work and decision-making is the responsibility of the project manager as he faces all the complexity and critical phases of this critical phase and deals with all its related processes which consist of defining the project tasks and assigning the employees responsible for each task and then preparing project plans. In this research we dealt with project scheduling using the smart dolphin swarm algorithm, the project scheduling process is essential in the software engineering industry nowadays and the possibility of finding good solutions which saves the software company a lot of time and money.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Baraa S. Mostafa

Department of Software Engineering

Universiti of Mosul (UOM)

Collede of Computer Science and Mathematic, Iraq

Email: baraa_sami@uomosul.edu.iq

1. INTRODUCTION

Software project scheduling problem (SPSP) is a critical process of assigning tasks to employees in a project so that completion time and cost are reduced. It differs from the well-known resource-constrained project scheduling problem (RCPS) in that there are two reduction goals while the RCPS only has one (time to completion), and also in SPSP the employees are the only source, each with a set of skills and a salary, but in RCPS there are many resources in quantitative quantities [1]. SPSP has a direct impact on the success of the project, it is the project management activity in which the project manager is responsible for and he must use different techniques and methodologies to manage the personnel and the tasks of the software project [2]. Thus, the process of scheduling software engineering projects is an important process that works to allocate employees to work on the tasks of software projects [3], as there are many schedules for a single project [4], but they differ in time and cost and the best schedule must be chosen and the project completed in the least possible time and cost, which increases. The complexity of the issue is the conflict of these objectives (project completion time and project cost) [5], and this increases the difficulty of finding the ideal solution. Therefore, there are two different techniques to solve this technical issue, the first is the use of the full examination method that examines all possible distributions or allocation. The technique will waste time, the second technique is the incomplete examination technique that examines only a part of the possible distributions or allocation in order to find the solution closest to the ideal solution in an acceptable time and cost less [6].

This technique includes heuristics and metaheuristics, the main distinction between inference methods and metaheuristics is that inference are methods designed to solve a specific problem only while metaheuristics can be adaptable to solve any improvement problem, and their research does not depend on the properties of the problem, therefore, it becomes intuitive to choose methods above metaheuristics, such as algorithms based on the intelligence of the swarm, which find the closest solution to the optimal solution in record time compared to the traditional methods [7]. The difficulty of this problem has generated much interest to solve scheduling problem starting with implementation the maximum and minimum ant algorithm with excessive cubic frame to solve the SPSP problem and compared its results with the results of other algorithms and it achieved the best results when the data were simple and uncomplicated [8], differential evolution algorithm and compared its results with the genetic algorithm in the SPSP issue [9], and used the differential evolution method with a new breakthrough technology in solving the SPSP problem and tested the algorithm using randomly generated data and compared its results with other algorithms and it reached the best results in a shorter time than the rest of the algorithms [10]. Using an evolutionary hyper-inference algorithm to solve the SPSP problem [11], luminous butterflies algorithm to solve the SPSP problem. They compared its results with the results of the genetic algorithm and the ant colony algorithm The luminous butterflies algorithm achieved the best results for small data [12]. In this research the smart dolphin swarm algorithm was used [13]. Inspired by the dynamic and living characteristics of the flock of dolphins in nature, in order to solve the issue of scheduling the project tasks in the least time and cost [14]. The rest of the paper is organized as follows: the second section describes the software project scheduling problem SPSP, the third section introduces the dolphin swarm algorithm (DSA), the fourth section introduces the proposed work, the fifth section presents result, the last section is a conclusion.

2. SOFTWARE PROJECT SCHEDULING PROBLEM (SPSP)

SPSP is defined as the process in which a project manager assigns project tasks to the employees who can work on it, and for the manager to do that, he must first obtain the following information [15]:

- a. Project tasks (Tasks): Tasks represent operations that must be executed in a sequential manner [16], taking into account the precedence of these tasks and the dependencies between them in order for the project to be completed. One task is known as (t_j) and j represent the values from (1) to (T) that represent the total number of project tasks. Each task possesses a set of skills known as t_j^{skills} , which must be available in the employee or group of employees responsible for the implementation of this task, as well as the effort required to implement it (Effort), which is known as t_j^{effort} [11]
- b. Task Precedence Graph: The task emergence diagram, known as (TPG), is used to represent the precedence of tasks and the dependencies between them.
- c. Employees: Employees are the most important source for software companies and they are the only source for SPSP. The employee is known as (e_i) and i represents the values from (1) to (E) that represent the total number of employees working on the project. Every employee possesses a set of skills e_i^{skills} .

3. DOLPHIN SWARM ALGORITHM (DSA)

The DSA has many stages, such as predation, searching, reception, calling, and predation, and it includes the predatory process of dolphins [17], and these habits and characteristics help the dolphins achieve their goal during the predation process. The dolphin swarm algorithm is consistent with notions of swarm intelligence [18], but differs from traditional swarm intelligence algorithms [19]. Depending on the intelligence of the swarm, a specific number of dolphins are needed to simulate the life habits and biological characteristics indicated in the actual predatory process of the dolphins, Figure 1 shown the flowchart of the dolphin swarm algorithm [11], and DSA can be divided into five stages as follows [20]:

- Initiation stage : initial dolphin swarm spawn evenly and randomly, $Dol_i = [x_1, x_2, \dots, x_D] T$ ($i = 1, 2, \dots, N$) Where N is the number of dolphins. x_j represents a component related to each of the dimensions to be improved. For every dolphin, there are two identical variants:
 - a) L_i is the perfect solution that Doli found at one time.
 - b) k_i Neighborhood is the best solution.
 Where $i = 1, 2, \dots, N$. After configuring the value of the dolphins, calculate the fitness of each of the dolphins and obtain a Fit_k . $Fit_k = \{Fit_{k,1}, Fit_{k,2}, \dots, Fit_{k,N}\}$ [21]
- Search stage : In this stage, each of the dolphins performs a search for their surrounding area by making sounds in random M directions. Likewise, the sound can be represented in the following way: $V_i = [v_1, v_2, \dots, v_D] T$, ($i = 1, 2, \dots, M$) in the current search, M stands for number of sounds and v_j , ($V_i = [v_1, v_2, \dots, v_D] T$, ($i = 1, 2, \dots, M$) in the current search, M stands for number of sounds and V_j , ($j = 1, 2, \dots, D$)

represent the component of each of the dimensions that are the sound characteristic of the direction. Moreover, the sounds satisfy $\|V_i\| = \text{velocity}$, ($i = 1, 2, \dots, M$), The "velocity" represents a constant equivalent to the sound characteristic of the velocity. In order to prevent the dolphins from getting stuck at this stage, the T_i : is maximum search time is set. At the maximum search time T_i , the voice of V_j which Dol_i , ($i = 1, 2, \dots, N$).

It results in the time when it will search for a new solution X_{ijt} , which can be expressed as:

$$X_{ijt} = Dol_i + v_i * t \tag{1}$$

For the new solution X_{ijt} that Dol_i is getting, the fitness of E_{ijt} can be calculated using the following formula:

$$E_{ijt} = Fit(X_{ijt}) \tag{2}$$

If ($E_{iab} = \min_{j=1,2,\dots,M; t=1,2,\dots,T1} E_{ijt}$). In this case, the individual optimal L_i of Dol_i solution will be determined:

$$L_i = X_{iab} \tag{3}$$

If ($\text{Fitness}(L_i) < \text{Fitness}(K_i)$). K_i is replaced by L_i ; Otherwise, in order not to change. After each Dol_i update ($i = 1, 2, \dots, N$) their L_i and K_i (in case they can be updated), the DSA goes into the call stage [22].

- Call stage In this stage, each of the dolphins produces sounds to inform the other dolphins of the results in the research stage, which includes whether a more adequate solution has been obtained and the location of that best solution [23]. ATS matrix where $TS_{i,j}$ represents (the rest of the time for the audio to travel from Dol_j to Dol_i and requires updating in the following way: for K_i, K_j and $TS_{i,j}$.

$$\text{If } (\text{fitness}(k_i) < \text{fitness}(k_j) \text{ and } TS_{i,j} = \left\lceil \frac{DD_{ij}}{A * \text{speed}} \right\rceil \text{ Then } TS_{i,j} = \left\lceil \frac{DD_{ij}}{A * \text{speed}} \right\rceil$$

Otherwise, $TS_{i,j}$ remains its value. Where ($i = 1, 2, \dots, N; j = 1, 2, \dots, N$) and Dol_j .

$$DD_{ij} = ||Dol_i - Dol_j||, i, j = 1, 2, \dots, N, i \neq j \tag{4}$$

Velocity is a constant equivalent to the sound property of velocity. A symbolizes a constant representing the acceleration capable of making sounds travel at a higher speed in case of very low speed, and then TS_{ij} undergoes an update based on (4) [23].

- The receiving stage in the DSA, the exchange process (which includes the communication and the receiving phases) will be maintained with TS , in case the DSA enters the receiving phase, each TS_{ij} term = $1, 2, \dots, n; j = 1, 2, \dots, N$), then TS_{ij} decreases by 1 to indicate that the sounds propagate over one unit of time. In this case, DSA requires verification of each TS_{ij} term in an array:

$$\text{If } (TS_{ij} = 0) \tag{5}$$

This means that the sound that is sent from Dol_j to Dol_i can be obtained by Dol_i , in which case there is a need to replace TS_{ij} with a new time term referred to as "maximum transmission time" (T_2), to indicate that the equivalent sound has been received. Make a comparison between K_i and K_j :

$$\text{If } (\text{fitness}(k_i) > \text{fitness}(k_j)) \tag{6}$$

In this case, then K_i will be replaced by K_j ; Otherwise, K_i remains unchanged. After each term in the matrix a TS_{ij} which satisfies in (5) is treated, and the DSA initiates the predation phase.

- Predation stage At this stage, each of the dolphins is required to calculate the radius surrounding the dolphin R_2 , and determine the distance between the optimum solution for the dolphin neighborhood and its position after the predation stage based on the available data, and after that, he obtains a new position. For each of the dolphins, the following is calculated

a) Distance DK_i :

$$DK_i = ||Dol_i - k_i||, i = 1, 2, \dots, N \tag{7}$$

b) DK_l Distance:

$$DKLi = || l_i - k_i ||, i = 1, 2, \dots N \tag{8}$$

R1 represents the search radius, and represents the maximum search phase range, and it can be calculated based on the following equation:

$$R1 = T1 * velocity \tag{9}$$

In general, the calculation of the surrounding radius R2 and the updating of the dolphin position should be discussed in three cases:

$$a) \begin{cases} \text{if } (DKi \leq Ri) \\ \text{then } R2 = (1 - \frac{2}{e})DKi \end{cases} \tag{10}$$

$$newDol_i = K_i + \frac{Dol_i - K_i}{Dk_i} R_2 \tag{11}$$

$$b) \begin{cases} \text{if } Dki > Ri \text{ and } DKi \geq DKLi \\ \text{then } R2 = (1 - \frac{\frac{-DKi}{Fitness(ki)} - \frac{DKLi - DKi}{Fitness(Li)}}{e.Dki \cdot \frac{1}{Fitness(Ki)}})DKi \end{cases} \tag{12}$$

$$newDol_i = K_i + \frac{random}{|random|} R_2 \tag{13}$$

$$c) \begin{cases} \text{if } (Dki < DKLi) \\ \text{then } R2 = (1 - \frac{\frac{DKi}{Fitness(ki)} - \frac{DKLi - DKi}{Fitness(Li)}}{e.Dki \cdot \frac{1}{Fitness(Ki)}})DKi \end{cases} \tag{14}$$

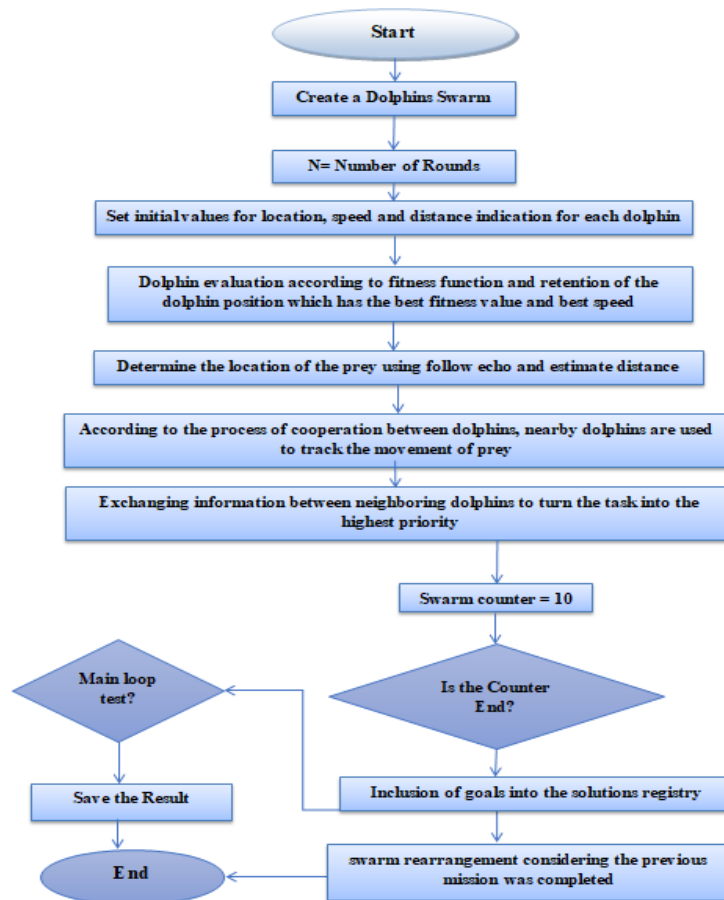


Figure 1. Flowchart dolphin swarm algorithm

Calculate newDol_i as (13), where e is a constant greater than 2. After Dol_i moves to the newDol_i position, newDol_i is compared to K_i in terms of fitness. Fitness (newDol_i) < Fitness (K_i). Then K_i is placed by newDol_i; Otherwise, in order not to change. After all Dol_i (i = 1, 2, ... , N) updates their locations and K_i (in the case it might update), if specified Dol_i meets the ending condition. If the ending condition is met, the DSA begins the termination phase. Else, the DSA begins the search phase again.

4. PROPOSED WORK

There are many challenges facing researchers in software project management issues, especially when conducting research on planning and scheduling, one of them is the lack of real data for the planning and scheduling phase of the project as the software companies reserve information related to employees, their salaries and their skills and do not provide any information on how to divide and distribute the work or any information related to management and leadership within the work team. Researchers have tried [24] to provide data For real projects in cooperation with a Jordanian software company, but that information is still few and limited, so researchers resort to using randomly generated data that simulate real projects when conducting research, and despite the good results achieved by the techniques and algorithms used, they are still not ready for use in tools and commercial applications [25].

The data used in this study are randomly generated data that simulate real software projects. It is the only source available to researchers for the purpose of studying the multi-purpose SPSP issue, which can be obtained from [26], This data consists of 36 groups, each group representing one project. These groups share some general characteristics and differ in other characteristics, as all groups possess 10 different skills available in the project staff, and each employee possesses 6-7 random skills out of a total of ten skills, as well as each task possesses 6-7 random skills out of the ten skills that must be provided. I have a group of employees responsible for carrying out this task [6]. These groups differ in the number of tasks and the number of employees in order to study the effect of the increase and decrease of these numbers on the resulting solutions and their accuracy. Therefore, these groups have 6 different numbers for the project tasks which are (16, 32, 64, 128, 256, 512) and 6 different numbers of employees (8, 16, 32, 64, 128, 256) [26].

Each group of data represents an independent project from the software projects. Therefore, each group has different dependencies and precedents between its tasks, meaning that each group has a different scheme from the task emergence schemes (TPG). Table 1 shows the similar and different characteristics of the groups while the Table 2 shows General description of each group of data.

Table 1. The similar and different characteristics of the data sets

Traits	Description
The total number of groups	36 groups
The total number of skills	10 different skills
Number of skills for each task	6-7 different skills
The number of skills per employee	6-7 different skills
The total number of tasks	Equals one of the following values (16, 32, 64, 128, 256 ,512)
The amount of effort	Each task has its own random voltage value
The total number of employees	Equals one of the following values (8, 16, 32, 64, 128, 256)
The amount of the salary	Each employee has their own random salary value
Task appearance diagram	Different random scheme for each project

Table 2. General description of each group of data

Dataset	General Description
16*8	consists of 16 task and 8 employees
16*16	consists of 16 task and 16 employees
16*32	consists of 16 task and 32 employees
16*64	consists of 16 task and 64 employees
16*128	consists of 16 task and 128 employees
16*256	consists of 16 task and 256 employees
32*8	consists of 32 task and 8 employees
32*16	consists of 32 task and 16 employees
32*32	consists of 32 task and 32 employees
256*256	consists of 256 task and 256 employees

The data collection process takes place according to the project information and its scheduling mechanism:

- Project tasks (Tasks): The tasks represent the operations that must be executed in a sequential manner, taking into account the precedence of these tasks and the dependencies between them in order for the

project to be completed. One task is known as (t_j) and j represents the values from (1) to (T) that represent the total number of project tasks. Each task possesses a set of skills known as (t_j^{skills}) which must be available in the employee or group of employees responsible for carrying out this task, and also possesses the effort required to implement it (Effort), which is known as (t_j^{effort}) [19].

- Task precedence graph: The task precedence graph, known as (TPG) is used to represent the precedence of tasks and the dependencies between them. It is a non-directed pie chart ($G(V, A)$) consisting of the top group ($(V = \{t_1, t_2, \dots, t_T\})$ vertex set) which expresses the project tasks and the arc set ((A) arc set) which expresses the dependency between the tasks as $((t_i, t_j) \in A)$ means that the task (t_i, t_j) It cannot begin to execute until after the task (t_i) is completed first [13].
- Employees: Employees are the most important source for software companies and they are the only source for SPSP. The employee is known as (e_i) and i represents the values from (1) to (E) that represent the total number of employees working on the project. Each employee possesses a set of skills (e_i^{skills}), each skill is known as (s_i) and i represents the values from (1) to (S) which represent the total number of skills of the employee, while (S_k) represents the number. The various skills available to the employees, one of the employees can possess two different skills (web design, programming in java) out of 6 different skills that the employees possess (designing websites, programming in java, team leadership, database management [26], programming in Matlab language, design using unified modeling language (UML)), and the skills of a single employee are considered a subset of the overall skill set ($\in S_k e_i^{\text{skills}}$). The employee has his own salary, also known as (e_i^{salary}) and is represented by a unit of money, as well as the value of the maximum time spent the employee to work in the project (maximum dedication), which is known as (e_i^{maxded}) and is represented by a unit of time, and this value is assumed to be equal to one [24].

The solution to the software project scheduling problem is a matrix X of size ($E \times T$) where ($X = x_{ij}$) the column represents the project tasks arranged according to precedence of implementation and the row represents the employees working on the project and the (x_{ij}) values represent a percentage of the upper limit of the time that the employee spends it in carrying out the task assigned to him, if ($x_{ij} = 0$) then this means that the employee (e_i) is not responsible for the task (t_j), and x_{ij} if ($=1$) then this means that the employee (e_i) devotes all his time to the task (t_j), but if it is ($0 < x_{ij} < 1$) then this means that the employee (e_i) allocates (x_{ij}) of his time to carry out the task (t_j) [27]. The above can be shown in Table 3.

Table 3. A summary of the issue of scheduling the software project

Description	Symbol
The total number of project tasks	T
Task number j in the project	t_j
The skill set for task t_j	t_j^{skills}
The effort required to implement the task t_j	t_j^{effort}
Task appearance diagram	TGP
Summit group, $V = \{t_1, t_2, \dots, t_T\}$	V
The bracket set, $(t_i, t_j) \in A$ where task t_i precedes task t_j by execution	A
The total number of employees in the project	E
Employee number i in the project	e_i
Employee salary e_i	e_i^{salary}
Employee Skillset e_i	e_i^{skills}
The total number of different skills available in the project	SK
The maximum time that an employee dedicates (e_i) to work on the project	e_i^{maxded}
Matrix solution by volume ($E \times T$)	X
Proportion of work done by employee e_i to perform task t_j	x_{ij}

5. RESULTS AND DISCUSSION

The two algorithms used in this message were implemented using (Matlab R2017a) language and on a laptop that is running on (Windows 10 Pro) system with (Core (TM) i5-4200U CPU 160GHz-2.30GHz) and has an external memory size (4 GB). As for the variables used by both algorithms, they can be shown in Table 4.

Table 4. The values of the variables of the DSA algorithms

Variables	DSA
Behaviors	The actions of predation, swarming, catching, and movement
Distance of vision	1.5
Step	0.01
The number of dolphins in the swarm	20
The size of the log of uncontrolled solutions	20
External log size	100
The number of sessions	1000
Number of times of implementations	10

After using the DSA algorithm in solving the SPSP problem, the following results were obtained:

- Execution time results: Table 5 shows the average execution time per second of the algorithm for all data sets. The execution time results obtained using the dolphin algorithm were compared with the source [28] that used the Artificial fish swarm algorithm-one behavior (AFSA-1b) and artificial fish swarm algorithm based on global (GAFSA) in the scheduling process. The Figure 2 shows the Average execution time in second between the DSA algorithm and the GAFSA, AFSA-1b algorithms.

Table 5. Average execution time in second

Data	DSA	AFSA-1b	GAFSA	Data	DSA	AFSA-1b	GAFSA
16*8	1.537	1.743	6.133	128*8	6.770	8.479	37.407
16*16	1.621	1.791	6.812	128*16	6.005	7.150	26.018
16*32	1.728	2.427	1014	128*32	9.873	11.535	40.683
16*64	2.004	3.179	13.364	128*64	19.894	22.234	81.682
16*128	3.502	5.211	8.805	128*128	28.689	43.809	173.293
16*256	5.886	9.678	32.383	128*256	55.984	88.986	306.882
32*8	1.680	2.616	10.148	256*8	9.509	15.916	67.5956
32*16	1.853	2.963	11.660	256*16	11.894	22.943	100.193
32*32	2.627	4.196	17.514	256*32	18.552	32.529	136.628
32*64	3.556	5.698	24.133	256*64	43.472	56.922	209.057
32*128	5.564	10.694	43.896	256*128	68.370	97.271	380.603
32*256	10.483	19.680	70.608	256*256	105.122	194.133	752.299
64*8	2.108	4.293	18.470	512*8	25.53	44.399	211.408
64*16	4.561	5.041	20.563	512*16	41.621	65.938	313.128
64*32	5.596	7.976	36.108	512*32	56.728	71.861	315.462
64*64	9.767	11.777	38.484	512*64	102.004	127.558	543.767
64*128	18.773	21.337	61.435	512*128	169.502	205.791	751.681
64*256	36.823	40.783	175.746	512*256	244.886	413.649	1503.57

- Diversity of results: The diversity of results is still the most important measure of performance evaluation. Therefore, the DSA algorithm was able to solve the SPSP issue in record time. Table 6 shows the mean HV value of DSA algorithm, solutions with an HV index value were shown for most datasets. HV value was compared with Resource [13] which contains 8 algorithms used in the scheduling process, as well as with AFSA-1b,GAFSA [28] shown in Table 7.

Table 6. The mean value of the oversize index

Data	DSA algorithm	Data	DSA algorithm	Data	DSA algorithm	Data	DSA algorithm
16*8	0.266	32*64	0.169	128*8	0.137	256*64	0.075
16*16	0.270	32*128	0.292	128*16	0.163	256*128	0.065
16*32	0.364	32*256	0.175	128*32	0.071	256*256	0.124
16*64	0.195	64*8	0.142	128*64	0.171	512*8	0.025
16*128	0.227	64*16	0.162	128*128	0.093	512*16	0.048
16*256	0.198	64*32	0.148	128*256	0.139	512*32	0.039
32*8	0.131	64*64	0.181	256*8	0.098	512*64	0.048
32*16	0.287	64*128	0.106	256*16	0.132	512*128	0.057
32*32	0.855	64*256	0.158	256*32	0.067	512*256	0.045

Table 7. The Comparison with value HV

Instance	NSGA- 	SPEA2	PAES	DEPT	MO- FA	MOAB C	MOCeII	GDE3	AFS A-1b	GAFSA	DSA
i64-8	0.469	0.3	0.8	0.0	0.1	0.0	0.5	0.5	0.07	0.0	0.1
		45	11	86	45	25	26	28	85	42	42
i64-16	0.219	0.1	0.9	0.0	0.3	0.1	0.2	0.3	0.07	0.0	0.1
		54	61	24	41	73	64	22	25	92	62
i64-32	0.065	0.0	0.7	0	0.3	0.0	0.0	0.2	0.09	0.0	0.1
		31	98		12	11	59	39	2	485	48
i64-64	0.074	0.0	0.8	0	0.0	0	0.0	0.2	0.10	0.0	0.1
		46	72		29		81	58	35	81	81
i64-128	0.026	0.0	0.7	0.0	0	0	0.0	0.1	0.09	0.0	0.1
		12	37	05			23	57	75	66	06
i64-256	0	0	0.6	0	0	0	0.0	0.1	0.12	0.1	0.1
			19				07	5	15	185	58

6. CONCLUSION

When previewing the previous results, the following can be observed; the dolphin algorithm was able to find results in the dataset (32*256) during the execution time (10.483) seconds, which is a relatively short time. Its new format is more suitable for use in commercial applications and software tools compared with the two algorithms (AFSA-1b GAFSA) that took (19.680, 70.608) respectively with the same data set. The dolphin algorithm proved that it is able to find various solutions where the mean values of the HV index in the dataset (64*64) is 0.181. It gave good diversity of solutions compared to the algorithms used in the scheduling process.

REFERENCES

- [1] Pressman, R.S., "Software engineering: a practitioner's approach," *McGraw Hill*, 2010.
- [2] Amiri, M. and Barbin, J.P., "New approach for solving software project scheduling problem using differential evolution algorithm," *International Journal in Foundations of Computer Science & Technology (IJFCST)*, vol. 5, no. 1, pp.1-9, 2015, doi: 10.5121/ijfcst.2015.5101.
- [3] Azizi, R., "Empirical study of artificial fish swarm algorithm," arXiv preprint arXiv:1405.4138, vol.3, no. 1, pp.1-7, 2014.
- [4] Sebt, M. H., Afshar, M. R., and Alipouri, Y. J. E. O., "Hybridization of genetic algorithm and fully informed particle swarm for solving the multi-mode resource-constrained project scheduling problem," *Engineering Optimization*, vol. 49, no. 3, pp. 513-530, 2017, doi: 10.1080/0305215X.2016.1197610.
- [5] Xiao, J., Gao, M. L., and Huang, M. M., "Empirical Study of Multi-objective Ant Colony Optimization to Software Project Scheduling Problems," In *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, 2015, pp. 759-766, doi: 10.1145/2739480.2754702.
- [6] Koulinas, G., Kotsikas, L., and Anagnostopoulos, K., "A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem," *Information Sciences*, vol. 277, pp. 680-693, 2014, doi: 10.1016/j.ins.2014.02.155.
- [7] Neshat, M., Sepidnam, G., Sargolzaei, M., and Toosi, A. N., "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artificial Intelligence Review*, vol. 42, no.4, pp. 965-997, 2014, doi: 10.1007/s10462-012-9342-2.
- [8] Crawford, B., Soto, R., Johnson, F., Monfroy, E. and Paredes, F., "A max-min ant system algorithm to solve the software project scheduling problem," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6634-6645, 2014, doi: 10.1016/j.eswa.2014.05.003.
- [9] Allan Vinicius Rezende, Leila Silva, André Britto, Rodrigo Amaral, "Software Project Scheduling Problem in the Context of Search-Based Software Engineering: A Systematic Review", *Journal of Systems and Software*, Vol 155, Pages 43-56, 2019.
- [10] Biju, A.C., Victoire, T. and Mohanasundaram, K., "An improved differential evolution solution for software project scheduling problem," *The Scientific World Journal*, vol, pp.1-9, 2015, doi: 10.1155/2015/232193.
- [11] Wu, X., Consoli, P., Minku, L., Ochoa, G. and Yao, X., "An evolutionary hyper-heuristic for the software project scheduling problem," *International Conference on Parallel Problem Solving from Nature*, pp.37-47, 2016, doi: 10.1007/978-3-319-45823-6_4.
- [12] Crawford, B., Soto, R., Johnson, F., Valencia, C. and Paredes, F., "Firefly Algorithm to Solve a Project Scheduling Problem," *Artificial Intelligence Perspectives in Intelligent Systems*, pp. 449-458, 2016, doi: 10.1007/978-3-319-33625-1_40.
- [13] Luna, F., González-Álvarez, D.L., Chicano, F. and Vega-Rodríguez, M.A., "The software project scheduling problem: A scalability analysis of multi-objective metaheuristics," *Applied Soft Computing*, vol. 15, pp.136-148, 2014, doi: 10.1016/j.asoc.2013.10.015.
- [14] Wu, T. Q., Yao, M., and Yang, J. H., "Dolphin swarm algorithm," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no.8, pp. 717-729, Aug. 2016, doi: 10.1631/FITEE.1500287.

- [15] F. Habibi, F. Barzinpour, S. J. Sadjadi., "Resource-constrained project scheduling problem: review of past and recent developments," *Journal of Project Management*, vol. 3, no. 2, pp.55–88, 2018, doi: 10.5267/j.jpm.2018.1.005.
- [16] Crawford, B., Soto, R., Johnson, F., Misra, S. and Paredes, F., "The use of metaheuristics to software project scheduling problem," In *International Conference on Computational Science and Its Applications*, 2014, pp. 215-226, doi: 10.1007/978-3-319-09156-3_16.
- [17] Amit Kumar, Sanjay Singla, Meenakshi Sharma, and Shakti Kundu., "Moth-Dolphin Optimization Algorithm: A Nature Inspired Technique," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 12, 2019, doi: 10.35940/ijitee.L3014.1081219.
- [18] Binitha, S. and Sathya, S.S., "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering*, vol. 2, no. 2, pp.137-151, 2012.
- [19] Minku, L.L., Sudholt, D. and Yao, X., "Improved evolutionary algorithm design for the project scheduling problem based on runtime analysis," *IEEE Transactions on Software Engineering*, vol. 40, no. 1, pp.83-102, 2014 doi: 10.1109/TSE.2013.52.
- [20] Fadhil, A. A., Alsarraj, R. G., and Altaie, A. M., "Software cost estimation based on dolphin algorithm," *IEEE Access*, vol.8, pp 75279-75287, 2020, doi: 10.1109/ACCESS.2020.2988867.
- [21] W. Qiao and Z. Yang, "Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function optimization problems," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2931910.
- [22] K. Lenin, B. R. Reddy, and M. S. Kalavathi, "Dolphin echolocation algorithm for solving optimal reactive power dispatch problem," *International Journal of Computer (IJC)*, vol. 12, no. 1, pp. 1–15, 2014, doi: 10.11648/j.ijcpe.20140301.11.
- [23] Qiao, W., and Yang, Z., "An Improved Dolphin Swarm Algorithm Based on Kernel Fuzzy C-Means in the Application of Solving the Optimal Problems of Large-Scale Function," *IEEE Access*, vol. 8, pp. 2073-2089, 2020, doi: 10.1109/ACCESS.2019.2958456.
- [24] W. Yong, W. Tao, Z. Cheng-Zhi, and H. Hua-Juan, "A new stochastic optimization approach-Dolphin swarm optimization algorithm," *International Journal of Computational Intelligence and Applications*, vol. 15, no. 2, 2016, doi: 10.1142/S1469026816500115.
- [25] Al Khatib, S.M. and Noppen, J., "Benchmarking and Comparison of Software Project Human Resource Allocation Optimization Approaches," *ACM SIGSOFT Software Engineering Notes*, vol. 41, no. 6, 2017, pp.1-6. doi: 10.1145/3011286.3011305.
- [26] Ferrucci F., Harman M. and Sarro F., "Search-Based Software Project Management," *Software Project Management in a Changing World*, pp. 373-399, 2014, doi: 10.1007%2F978-3-642-55035-5_15.
- [27] Alba, E. and Chicano, J.F., "Software project management with Gas," *Information Sciences*, vol. 177, no. 11, pp.2380-2401, 2007, doi: 10.1016/j.ins.2006.12.020.
- [28] Sarah E. Almshadany, Laheeb Ibrahim, "Using multi-objective artificial fish swarm algorithm to solve the software project scheduling problem," Master's thesis, in Software Engineering, Collage of computer of mathematics, University Of Mosul, Iraq, 2019.