

## Using Artificial Bat Swarm Algorithm In Scheduling Problem

Firdews A.Alsalman<sup>1</sup>, Baraa S.Mostafa<sup>2</sup>

<sup>1</sup>University of Mosul (UOM), College of Computer Science and Mathematics

<sup>2</sup>Department of software Engineering

[Firdewsalsalman@uomosul.edu.iq](mailto:Firdewsalsalman@uomosul.edu.iq), [baraa\\_sami@uomosul.edu.iq](mailto:baraa_sami@uomosul.edu.iq)

**Article History:** Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 4 June 2021

**Abstract:** Software Project scheduling problem is considered one of the basic and important processes in the software industry at the present time as software manufacturers strive to meet customer needs in terms of time and money factor and the ability to find good solutions where we can achieve 60% of project success if planned and make good use of human resources. And other resources The project manager is responsible for planning the resources and identifying the critical points facing the project in addition to defining the tasks assigned to the employees to implement them according to their skills, experience and the extent of relevance of each task to another. In this paper, project scheduling is presented using the intelligent bat swarm algorithm.

**Keywords:** Project scheduling, bat swarm algorithm, swarm intelligence algorithms, software engineering.

### 1. Introduction

The accuracy of planning a software project is dependent on the ability to define and adjust all tasks according to the skills of each employee and the ability to determine the effort for each task and the appropriate time to complete each task to achieve the goal to complete all tasks and schedule them in the correct and appropriate manner according to their specifications and the deadline for the delivery of the project, the problem of project planning and scheduling A very accurate process in terms of accurately defining tasks for project personnel depending on the principle of reducing the time and cost of the project, which are the two most important goals in scheduling a software engineering project and in the Software project scheduling problem process. Employees are the only source, and each of them has a skill and a salary [2]. There are many timesheets for a single project, but the difference between these schedules lies in time, cost and the best schedule that must be determined and completed according to the client's requirements. [3]. Bearing in mind that there are no conflicts in the executed tasks, which makes it more difficult to find the best scheduler (the optimal solution) [10] The real problem facing the scheduling process is in the conflict of these basic objectives, which are (project completion time and project cost) [5], and this in the nature of the case increases the difficulty of finding the ideal best solution. So , there are two different ways to solve this technical problem. The first method is by using the full scan method that examines each distributions or assignment[4]. It is possible, but one of the disadvantages of this technique will be wasted time. The second method is the incomplete examination method that relies on examining only part of the possible distributions or assignments in order to find the closest solution to the ideal solution in an acceptable time and at a lower cost. [6] The project manager is not able to experience all the possibilities of distributing the implementation of tasks to the employees and choosing the best solution that allows him to complete the project in the least time and cost. Therefore, researchers resort to using intelligent methods to find the closest solution to the best solution to help the project manager in this complex issue [11].

### 2. Significance Of The Study

When the company starts working on a new project, it tries to complete it in less time and less cost, but the reduction in time always means an increase in the cost, and the reverse is also true, and this is certainly reflected in planning project and scheduling, so the importance of this research lies in improving these issues by relying on the optimization of the goal The one or the multi-purpose optimization and despite the complexity caused by the multi-purpose optimization, this representation is closer to setting up projects in the real world where the intelligent bat algorithm was used for the purpose of improving the project scheduling process.

### 3. Review Of Related Studies

Several techniques have been used in order to achieve project scheduling, such as using In researchers **Enrique Alba, J. Francisco (2007)** proposed the issue of SPSP genetic algorithm to find a solution by relying on a data generator specially designed to obtain data that simulates real projects .These data were divided into five groups, each of which has specific characteristics, in order to test the algorithm, which in turn achieved results. Good in

some groups and bad results in others [3], **Chicano Francisco(2013)**, a researcher, with a group of researchers, tested eight algorithms above the multi-purpose heuristics to solve the issue of software project scheduling by using Pareto efficiency. The performance of the advanced Pareto archiving strategy algorithm was the merit and achieved higher quality results than other algorithms.[11], the researcher **Yao Xin (2014)**, along with a group of researchers, presented some improvements in solving the SPSP issue and presented an experimental study using an improved design of the evolutionary algorithm based on runtime analysis, and they found that the improvements were effective and increased the quality of the solution and made the rate of modification on the members of the generation correct by a percentage 100% [13], **Biju. C. A(2015)** with other researchers, used the method of differential evolution using a new breakthrough technology in solving the SPSP problem, and they tested the algorithm using randomly generated data and compared its results with other algorithms. It reached the best results in a shorter time than the rest of the algorithms [6].and the researcher **Minku Leandro and a group of researchers (2016)** proposed several methods in selecting transit and mutation processes when using an evolutionary hyper-inference algorithm to solve the SPSP problem and proved that the proposed algorithm can solve the problem efficiently [19].

#### 4. Objectives Of The Study

- Comparing the performance of the bat algorithm that proposed based on the principles of the Archived Pareto Evolutionary Strategy (PAES), which achieved very good results in solving the SPSP problem, and other algorithms used in solving the scheduling problem.
- Reducing costs and time in the project scheduling process

#### 5. Hypotheses Of The Study

- At least one employee must carry out the task and not leave any of the project tasks without being assigned to any of the employees
- The skills needed by the task must be available in the skill set of the personnel responsible for it.
- The amount of work done by the employee should not exceed the maximum time allotted to work on the project, that is, the employee should not do any additional work (overwork).

#### 6. Population And Sample

The data used in this research is a randomly generated data that simulates real software projects, which is the only source available to researchers for the purpose of studying the SPSP issue, and can be obtained from [17]. 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.

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, 8). 16, 32, 64, 128, 256), 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) illustrates the similar and different characteristics of the groups.

**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

The following is a description of the input matrices for the dataset (8 \* 1)

1. Matrix of tasks - skills: This matrix consists of 10 columns, each column represents a skill from the ten skills available in the project, and 16 rows, each row represents one of the project tasks. If the value between any row and column is equal to (1) then this means that the task needs the skill and if it is (0) then this means that the task does not need that skill.
2. The Estimation vector: This vector consists of (16) values, each of which represents the effort required to complete the task. Table (2) illustrates the task-skills matrix as well as the effort vector.

**Table 2:** Matrix of tasks-skills and effort vector for the dataset (8 \* 16)

	Skill 1	Skill 2	Skill 3	Skill 4	Skill 5	Skill 6	Skill 7	Skill 8	Skill 9	Skill 10	Effort
Task 1	1	1	0	1	0	1	1	0	1	1	12
Task 2	0	1	1	1	1	1	1	1	0	0	4
Task 3	1	1	1	0	1	0	0	1	1	1	19
Task 4	0	1	1	1	1	0	1	1	0	1	12
Task 5	1	1	1	0	0	1	1	1	0	1	5
Task 6	1	1	0	1	1	1	1	0	0	0	5
Task 7	1	1	0	1	0	0	1	1	1	1	9
Task 8	1	1	0	1	1	1	0	1	0	0	12
Task 9	1	0	1	1	1	0	0	1	1	0	4
Task 10	1	1	0	0	0	0	1	1	1	1	5
Task 11	0	1	1	0	1	0	1	1	1	1	7
Task 12	0	0	1	0	0	1	1	1	1	1	18
Task 13	0	0	0	1	1	0	1	1	1	1	22
Task 14	0	0	1	1	1	1	1	0	1	1	15
Task 15	1	0	1	1	1	0	0	1	1	1	9
Task 16	0	1	1	0	1	0	1	1	1	1	22

3. Personnel matrix - skills: which consists of (10) columns representing project skills and (8) rows representing project employees, if the value between any row and column is equal to (1) then this means that the employee possesses that skill, either if it is equal to (0) It means that he does not have that skill.
4. Salary Vector: It consists of (8) random values, each of which represents the salary of each employee. Table (3) illustrates the personnel matrix - skills and salary vector.

**Table 3:** Personnel-Skills Matrix and Salary Vector for Data Set (16 \* 8)

	Skill 1	Skill 2	Skill 3	Skill 4	Skill 5	Skill 6	Skill 7	Skill 8	Skill 9	Skill 10	Salary
Employee 1	1	1	1	0	1	1	1	0	0	1	9593.39
Employee 2	1	0	0	1	1	1	1	0	0	1	9161.667
Employee 3	0	1	0	1	1	0	1	1	1	1	9144.447
Employee 4	1	1	0	1	1	1	1	0	0	1	10733.75
Employee 5	0	1	0	1	1	1	1	1	1	0	9643.789
Employee 6	0	1	1	1	1	1	1	1	0	0	8788.155
Employee 7	1	1	1	0	1	0	1	0	1	1	10363.94
Employee 8	1	1	1	1	0	0	1	1	1	0	10645.98

5. The task emergence diagram matrix: It is a binary matrix that shows the precedence of implementation among the tasks and consists of two columns, the first representing the tasks that must be carried out first and the second column representing the tasks that are implemented after the first tasks. The first row of (1, 2) shows that task (1) precedes task (2) in implementation.

**6.1 Software Project Scheduler Problem Used in the Present Study**

The scheduling process is the process by which the project manager assigns tasks to the appropriate employees to work on, and in order to do so, he must obtain the following information [9]

- Project tasks (tasks): Tasks represent the processes that have been executed in a sequential fashion [10], taking into account the precedence of these tasks. One task is known as  $t_j$  and  $j$  represents the values from (1) to (T), which represent the total number of tasks. You possess an important set of skills that you want to be present in the employee or group of employees. Performing this task, the effort required to perform it (effort), which is known as  $t_j^{effort}$ .
- Task precedence chart: This uses a chart, known as a TPG, to represent the precedence of tasks [16].
- Employees: Employees are the single most important resource for software companies and at SPSP. The employee is known as (ei) and  $i$  represent the values from (1) to (E) that represent the total number of employees working on the project. Every employee possesses a set of skills [8]

SPSP has two objectives that must be improved during the solution which are project completion time and project cost.

A .The project completion time is calculated with the following steps [11] :

1. The value of ( $t_{jahr}$ ) is calculated for all project tasks and it is a value that represents the amount of work the employees put in to implement the task ( $t_j$ ) and this is done by adding all units of time ( $x_{ij}$ ) for that task using the following equation:

$$\dots\dots\dots (1)t_j^{ahr} = \sum_{i=1}^E x_{ij}$$

Where as:

$t_j^{ahr}$  = The amount of work the employee.  $x_{ij}$  = The percentage of work he does to get the job done.

2. The duration of each task is calculated ( $t_j^{dur}$ ) by the following equation:

$$\dots\dots\dots (2)t_j^{dur} = \frac{t_j^{effort}}{t_j^{ahr}}$$

Where as:

$t_j^{dur}$  = Duration of task execution,  $t_j^{effort}$  = Task-specific effort.,  $t_j^{ahr}$  = The amount of work the employees did to get the job done.

3.The start and end times for each task are calculated ( $t_j^{start}, t_j^{end}$ ) by the following two equations:

$$\dots\dots\dots (3)t_j^{start} = \begin{cases} 0 & \text{if } \nexists t_i, (t_i, t_j) \in A \\ \max_{t_i, (t_i, t_j) \in A} \{t_i^{end}\} & \text{otherwise} \end{cases}$$

$$\dots\dots\dots (4)t_j^{end} = t_j^{start} + t_j^{dur}$$

Where as:

$t_j^{start}$  = The start time of the task execution, which is equal to zero if the task is not preceded by another task in the execution or equal to the highest end time of the execution of the tasks that precede the current task in the execution.

$t_j^{end}$  = The end time for the task.  $t_j^{dur}$  = Duration of task execution.

4. The project completion time can now be calculated ( $p_{dur}$ ) Which equals the time the last task completed execution: $p_{dur} = \max_{j=1} \{t_j^{end}\}$  ..... (5)

Where as:

$p_{dur}$  =Project completion time ,  $t_j^{end}$  = The end time for the task.

B. As for the cost, it can be calculated in the following steps [17]:

1.The cost of implementing each task must first be calculated ( $t_j^{cost}$ ):

$$t_j^{cost} = \sum_{i=1}^E e_i^{salary} . x_{ij} . t_j^{dur} \dots\dots\dots (6)$$

Where as:

$t_j^{cost}$  = The cost of implementing the task.

$e_i^{salary}$  = The employee's salary.  $x_{ij}$  = The percentage of work the employee does to perform the task.

$t_j^{dur}$  = Duration of task execution.

2.The total cost of the project ( $p_{cost}$ ) is equal to the total cost of implementing all the tasks of the project [30]:

$$\dots\dots\dots (7) p_{cost} = \sum_{j=1}^T t_j^{cost}$$

Where as:

$p_{cost}$  = The total cost of the project,  $t_j^{cost}$  = The cost of implementing the task.

There are a set of limitations and limitations that must be taken into consideration when solving the SPSP question in order for the solution to be considered correct and for it to represent the real schedule of the software project, namely [8]:

1.The task must be carried out by at least one employee, and none of the project tasks should be left without being assigned to any of the employees.

$$t_j^{ahr} > 0 \forall j \in \{1, 2, \dots, T\} \dots\dots\dots (8)$$

Where as:  $t_j^{ahr}$  = The amount of work the employees to get the job done.

2. The skills needed by the task must be present in the skill set of the personnel responsible for it.

$$\dots\dots\dots (9) t_j^{skills} \subseteq \cup_{\{i|x_{ij} > 0\}} e_i^{skills} \forall j \in \{1, 2, \dots, T\}$$

Where as:  $t_j^{skills}$  = Task-specific skill set.  $e_i^{skills}$  = The employee's skill set.

3. The amount of work that the employee does should not exceed the maximum time allocated to work on the project, meaning that the employee does not do any additional work

$$\dots\dots\dots (10) e_i^{work}(\mathcal{T}) = \sum_{\{j|t_j^{start} \leq \mathcal{T} \leq t_j^{end}\}} x_{ij}$$

Whereas:

$e_i^{work}(\mathcal{T})$  = The total work an employee does on the project over time (T),  $x_{ij}$  = The percentage of work the employee does to perform the task.

If the employee's work amount is greater than the employee's maximum time ( $e_i^{work}(\mathcal{T}) > e_i^{maxded}$ ) in the time (T), This indicates the emergence of additional work in the schedule, which can be calculated by the following equation:

$$e_i^{over} = \int_{r=0}^{r=p^{dur}} ramp(e_i^{work}(\mathcal{T}) - e_i^{maxded}) d\mathcal{T} \dots\dots\dots (11)$$

Whereas :

$e_i^{over}$  = Overtime for the employee,  $e_i^{work}(\mathcal{T})$  = The total work an employee does on the project over time (T),  $e_i^{maxded}$  = The maximum time that an employee dedicates to work on the project.

$ramp$  = A function that can be defined by:

$$\dots\dots\dots (12) ramp(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

The amount of total overtime work on the project equals the sum of the employees' overtime values:

$$\dots\dots\dots (13) p_{over} = \sum_{i=1}^E e_i^{over}$$

Whereas:

$p_{over}$  = Total overtime work on the project,  $e_i^{over}$  = Overtime for the employee.

No additional work must appear in the schedule.

$$\dots\dots\dots (14) p_{over} = 0$$

This condition is the most difficult condition, and when additional work appears for any of the employees, then a coefficient must be used for the purpose of returning the employee's time value to be less or equal to the upper limit through the following equation [11]:

$$x_{ij} = \frac{x_{ij}}{\max_{i,T}\{e_i^{work}(\mathcal{T}) + \epsilon\}} \dots\dots\dots (15)$$

Whereas:

$\hat{x}_{ij}$  = The percentage of work that the employee does to implement the task after applying the correction factor.

$x_{ij}$  = The percentage of work that the employee does to implement the task before applying the correction factor.

$\epsilon$  = A value used to prevent computational errors ( $\epsilon = 0.00001$ ).

The use of the correction factor affects the end time of the project, as it is subjected to an increase as in the following equation:

$$\hat{p}_{dur} = p_{dur} + \max_{i,T} \{e_i^{work}(T) + \epsilon\} \dots\dots\dots (16)$$

Whereas:

$\hat{p}_{dur}$  = Project end time after applying the correction factor,  $p_{dur}$  = Project completion time before applying the correction factor.

As for the cost of the project, it is not affected by the correction factor:

$$\hat{p}_{cost} = p_{cost} \dots\dots\dots (17)$$

$\hat{p}_{cost}$  = The cost of the project after applying the correction factor.

$p_{cost}$  = The cost of the project before applying the correction factor.

**6.2 Intelligent Bat Swarm Algorithm (IBS)**

In 2010 a new optimization algorithm was introduced by Yang, which he named the intelligent Bat Swarm algorithm [1] based on the behaviour of the intelligence of the swarm and the inspiration from the observation of Microbat’s properties [20].

**Intelligent Bat Swarm Algorithm have three main properties that exist for microbat’s are used to build the structure of the main bat algorithm [5]**

- A) The echolocation trait: The majority of bat species use echolocation behaviour to locate and discover prey.
- B) Frequency characteristic: A fixed frequency  $f_{min}$  is transmitted by the microbes with a variable wavelength  $A$  and loudness of  $A_0$  in order to know the location of the prey.
- C) The quality of loudness: the loudness of its values varies from a large positive value  $A_0$  to a minimum constant value, represented by an  $A_{min}$ .

On the basis of Equation 20, the virtual movement of bats is regulated

$$f_i = f_{min} + (f_{max} - f_{min}) \beta \dots\dots\dots(18)$$

where  $f_i$  represents the frequency which is using by a bat which looking for the prey,  $f_{min}$  and  $f_{max}$  represent the minimum value and maximum value, respectively,  $\beta$  is an arbitrary vector, drawn from a uniform distribution, and  $\beta \in [0, 1]$ .

$$v_{it} = v_{it-1} + (x_{it} - x_{best}) f_i \dots\dots\dots (19)$$

$$x_{it} = x_{it-1} + v_{it} \dots\dots\dots (20)$$

The value of  $x_i$  represents  $i$ th bat location in the space of the solution, value of  $v_i$  is the bat’s velocity,  $t$  value represents the present iteration, and  $x_{Best}$  represents global near optimal solution which has obtained to the point, over the entire population [21]A new solution for the bat will be produced based on Eq. 21[7]:

$$x_{new} = x_{old} + \epsilon A_t \dots\dots\dots (21)$$

where  $\epsilon$  value represents an arbitrary number which  $\epsilon \in [-1, 1]$ , and  $A_t$  is the average of loudness for all the bats at the present time step. After the the update of the bats’ positions, pulse emission rate  $r_i$  and loudness  $A_t$  are updated, as well only in the case where the global near optimal solution is altered and the randomly produced number is  $< A_i$ .  $A_i$  and  $r_i$  updates are regulated with Eq. 22.

$$A_{it+1} = \alpha \cdot A_{it} \dots\dots\dots(22)$$

where  $\alpha$  is constants and  $\alpha = 0.9$  is Simply used for, Check the termination for determine whether going back to Eq. 20 or finished the algorithm and producing the near optimal solution [12], The Figure.1 represents the stages of Intelligent Bat Swarm Algorithm [14], and Figure.1 showing intelligent Bat Swarm Algorithm[17].

### 6.3 Modelling the solution and Result

The solution to the SPSP problem is a table represented by a matrix of size (E \* T), (E) represents the number of project employees and (T) represents the number of project tasks, and each point in that matrix expresses the value of time spent by the employee in carrying out the task, If the value is  $x_{ij} = 0$ , then it means the employee does not work on that task, but if it is a value  $1 > x_{ij} > 0$ , then this means that the employee works at a volume  $x_{ij}$  to carry out that task. But if the value of  $x_{ij} = 1$  means that the employee devotes all his time to the implementation of the task  $t_j$ . Each bat in the swarm represents a different schedule that is randomly generated at start of execution. Every artificial bat must meet the determinants of the SPSP question (which are verified by formulas (8), (9) and (10)) as no task should remain without any employee responsible for it. The group of employees responsible for the implementation of a specific task must possess all the skills that the task needs to be executed properly, and these two conditions can be easily achieved, but the difficulty lies in achieving the third condition, which is that the amount of time that the employee spends in carrying out his tasks does not exceed the upper limit of time. Allocated for work on the project. Table (4) represents a random artificial bat and shows the total amount of time allocated to work on the project.

Figure.1: intelligent Bat Swarm Algorithm

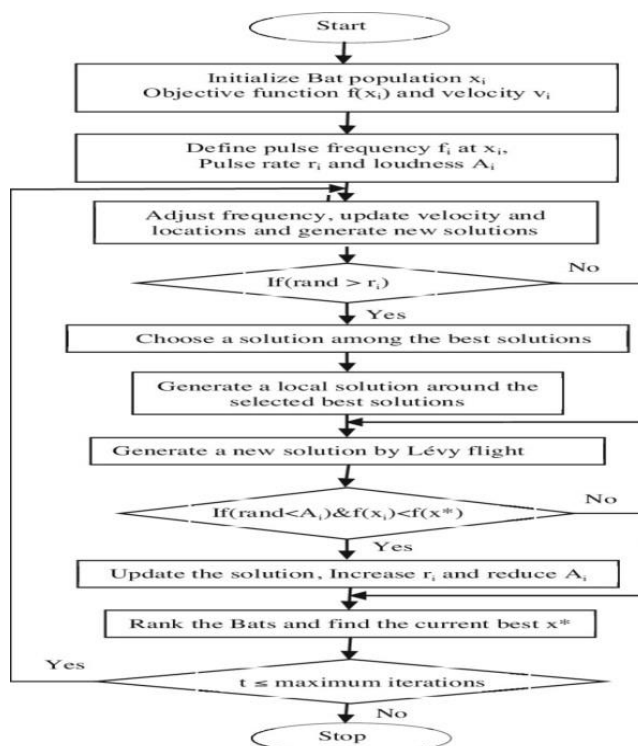


Table 4: represents a random artificial bat and shows the total amount of time allocated to work on the project.

	Task1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Total Time
Employee 1	0	0	0.324	0.157	0	0.221	0	0.324	1.026
Employee 2	0	0.765	0.194	0.219	0.325	0.281	0	0.126	1.910
Employee 3	0	0	0	0	0.165	0.344	0.421	0.764	1.694
Employee 4	0.243	0.015	0.173	0	0	0	0	0.632	1.063
Employee 5	0	0.351	0	0.625	0	0.402	0.974	0.410	2.762
Employee 6	0.281	0.254	0	0	0.014	0	0	0	0.549
Employee 7	0	0.310	0	0	0.204	0	0	0	0.528
Employee 8	0	0.142	0	0.036	0.912	0	0.347	0.042	1.479

If a work is assigned to one of the employees more than the employee's time, then this means that the employee is forced to work overtime (the overtime is calculated by equation (11)) and this violates the terms of the issue, so it must use the correction factor (it is calculated using the equation (15)). Which divides all the artificial bat

values into the highest time estimate found by the employees. Table (5) The artificial bat, after applying the correction factor to it, is now the correct solution to the SPSP problem.

**Table 5:** shows the total amount of time for each employee after applying the correction factor.

	Task1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Total Time
<b>Employee 1</b>	0	0	0.023	0.034	0	0.033	0	0.021	0.111
<b>Employee 2</b>	0	0.053	0.027	0.104	0.147	0.041	0	0.107	0.479
<b>Employee 3</b>	0	0	0	0	0.217	0.109	0.084	0	0.215
<b>Employee 4</b>	0.126	0.055	0.174	0	0	0	0	0	0.355
<b>Employee 5</b>	0	0.115	0	0.158	0	0.127	0.205	0.074	0.679
<b>Employee 6</b>	0.032	0.125	0	0	0.204	0	0	0	0.361
<b>Employee 7</b>	0	0.035	0	0	0.041	0	0	0	0.076
<b>Employee 8</b>	0	0.026	0.012	0	0.134	0	0.221	0.001	0.394

Now it is possible to calculate the objectives of the SPSP question (calculate the project completion time through equations (1) to (5) and calculate the project cost through equations (6) and (7)) for the artificial bat in the previous example, Table (6) represent the goal values, the first value represents the project completion time (Time) and the second value represents the project cost.

**Table 6:** represent the goal values.

Project time	Project cost
235.301	24677

Since the size of the artificial bat consisting of the problem data is very large (the smallest size of the artificial bat is  $(16 * 8)$  points, it becomes very difficult to implement the echolocation and frequency behaviour of the bat algorithm literally, which requires a lot of time and storage space as well as the resulting complexity About dealing with artificial bats that are large binary arrays (especially when calculating the value of the distance between the current bat and other bats in the swarm), Therefore, the echolocation behaviour was applied assuming the value of  $x_c$  is equal to the artificial bat that has the mean echo value .When the echo action was implemented, the value of  $x_j$  was considered as any artificial bat in the swarm that has an echo value greater than the echo value of the current bat.  $x_j$  becomes the bat with the second largest echo value in the swarm.In order to compare the performance of the intelligent bat swarm algorithm by the correction factor by noting the significant difference in the percentage of applying the correction factor with the source [13] that used AFSA-1b and GAFSA algorithms, where the intelligent bat algorithm needed to apply the parameter much less than the number of times that where the parameter has been applied before. Both algorithms and this reflects the ease of implementation of the bat algorithm as shown in Table (7).

**Table 7:** The percentage of applying the correction factor

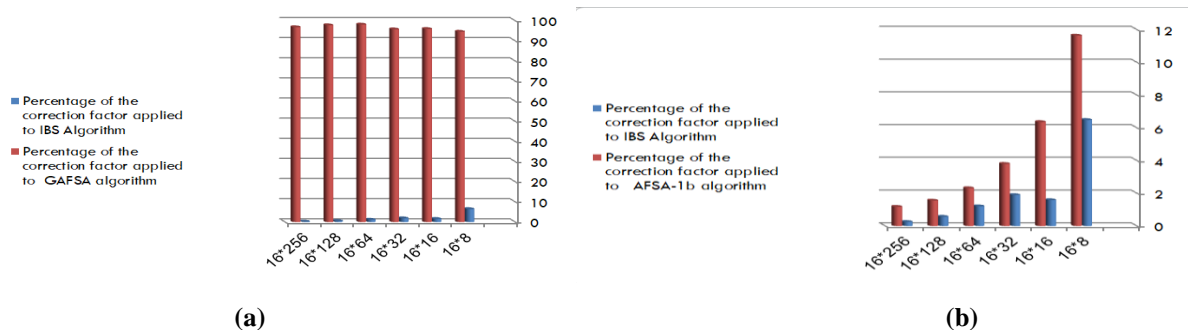
Data	IBS	GAFSA	AFSA-1b	Data	IBS	GAFSA	AFSA-1b
<b>8*16</b>	6.485	94.5965	11.63750	<b>16*128</b>	3.458	99.832	7.1435
<b>16*16</b>	1.586	95.9425	6.358	<b>32*128</b>	2.254	99.9375	3.9715
<b>32*16</b>	1.896	95.7705	3.7995	<b>64*128</b>	1.245	99.936	2.3835
<b>64*16</b>	1.210	98.1345	2.3145	<b>128*128</b>	0.128	99.923	1.581
<b>128*16</b>	0.569	97.81	1.558	<b>256*128</b>	0.048	97.692	1.133
<b>256*16</b>	0.253	96.8275	1.176	<b>8*256</b>	9.875	96.6615	14.3575
<b>8*32</b>	8.145	99.7755	13.5045	<b>16*256</b>	3.865	96.8295	7.158



<b>16*32</b>	2.859	96.527	6.7635	<b>32*256</b>	2.486	99.958	3.9965
<b>32*32</b>	2.893	96.328	3.873	<b>64*256</b>	0.896	96.458	2.354
<b>64*32</b>	1.230	97.8885	2.33	<b>128*256</b>	0.158	99.985	1.5815
<b>128*32</b>	0.879	98.2825	1.516	<b>256*256</b>	0.456	96.7775	1.197
<b>256*32</b>	0.125	97.8605	1.1725	<b>8*512</b>	6.798	96.4925	15.442
<b>8*64</b>	8.165	99.848	13.3925	<b>16*512</b>	4.586	96.4495	7.396
<b>16*64</b>	2.758	96.5285	6.8875	<b>32*512</b>	2.786	99.977	4.072
<b>32*64</b>	2.547	99.181	3.8965	<b>64*512</b>	0.896	99.982	2.414
<b>64*64</b>	1.324	96.479	2.402	<b>128*512</b>	0.345	96.5215	1.618
<b>128*64</b>	0.725	97.8575	1.5625	<b>256*512</b>	0.214	96.4395	1.171
<b>256*64</b>	0.048	97.587	1.113	<b>16*128</b>	3.458	99.832	7.1435
<b>8*128</b>	9.258	99.871	14.258	<b>32*128</b>	2.254	99.9375	3.9715

The comparison in percentage of the correction factor between Intelligent Bat swarm algorithm (IBSA) and the algorithms used in the source in the scheduling project software problem shown in figure .2

**Figure.2:** The Comparison in percentage of the correction factor between (a) (IBS) with GAFSA,(b) IBS with AFSA



## 7. Recommendations

After the bat algorithm achieved very good results in solving the SPSP problem, it becomes possible to suggest a set of axes that can become as subsequent recommendations, as follows:

- Study the performance of the intelligent bat swarm algorithm when changing the values of its variables and the effect of that change on the efficiency of the solution and the accuracy of the results, as well as finding the values of the ideal variables that achieve the best results for the SPSP issue.
- Studying the possibility of proposing a percentage of the update that is made on each artificial bat instead of updating only one of the points of the bat, and studying the effect of this increase on the accuracy and variety of results and the speed of implementation.
- Using the bat algorithm in solving other problems with multiple objectives, studying its performance and dealing with the characteristics of those problems.
- Hybridizing the bat swarm algorithm with other intelligent methods and trying to find effective formulas to solve complex problems
- There are still many important issues that are dealt with as issues of one goal, but this representation does not reflect the reality that rarely contains an issue that has a single goal. Therefore, the general formula for multi-objective issues can be used to convert issues with one goal into a more form. Realistic and closer to real world matters.

## 8. Conclusion

When comparing the performance of the bat algorithm with the performance of the AFSA-1b and GAFSA algorithms in solving the SPSP problem, the following conclusions are drawn:

The bat algorithm is easier to implement than the AFSA-1b algorithm and the GAFSA algorithm, as it needed to apply the correction factor of (1.586) when resolving the data set (16 \* 16). This indicates the limited change that the proposed algorithm makes to the synthetic bat. This also indicates The ease of these operations, while the AFSA-1b algorithm needed to apply the parameter of (6.358) and GAFSA algorithm is to apply the parameter of (95.9425) when solving the same dataset and this indicates the difficulty of implementation and the difficulty of the operations that are performed.

## References

1. A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, 2013, "Metaheuristic algorithms in modeling and optimization," in *Metaheuristic Applications in Structures and Infrastructures*, pp. 1\_24.
2. Al Khatib, S.M. and Noppen, J., 2017, "Benchmarking and Comparison of Software Project Human Resource Allocation Optimization Approaches". *ACM SIGSOFT Software Engineering Notes*, Vol. 41, No. 6, pp.1-6.
3. Alba, E. and Chicano, J.F., 2007, "Software project management with Gas". *Information Sciences*, Vol. 177, No. 11, pp.2380-2401.
4. Amiri, M. and Barbin, J.P., 2015, "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.
5. Anfal A. fadhil.Rasha G. alsarraj,Atica M.Altai, 2020, "Software cost estimation based on dolphin algorithm", *IEEE*, vol.8,pp 75279-75287,2020
6. Biju, A.C., Victoire, T. and Mohanasundaram, K., 2015, "An improved differential evolution solution for software project scheduling problem". *The Scientific World Journal*, Vol. 2015, pp.1-9
7. Crawford, B., Soto, R., Johnson, F., Misra, S. and Paredes, F, 2014, "The use of metaheuristics to software project scheduling problem". In *International Conference on Computational Science and Its Applications*, Springer, pp. 215-226.
8. Farhad Habibi, Farnaz, 2018 "Resource-constrained project scheduling problem: review of past and recent developments", Barzinpour and Seyed Jafar Sadjadi ", *Journal of Project Management*, pp.55–88,
9. Ge, H., Sun, L., Chen, X. and Liang, Y., 2016, "An Efficient Artificial Fish Swarm Model with Estimation of Distribution for Flexible Job Shop Scheduling". *International Journal of Computational Intelligence Systems*, Vol. 9, No. 5, pp.917-931.
10. Goodrich, M.T., Tamassia, R. and Goldwasser, M.H., 2014, "Data structures and algorithms in Java". John Wiley & Sons.
11. Luna, F., González-Álvarez, D.L., Chicano, F. and Vega-Rodríguez, M.A., 2014, "The software project scheduling problem: A scalability analysis of multi-objective metaheuristics". *Applied Soft Computing*, Vol. 15, pp.136-148.
12. M. Ibrahim, 2016, "A new model for software cost estimation using bat algorithm," *Int. J. Acad. Res. Comput. Eng.*, vol. 1, no. 1, pp. 53\_60.
13. Minku, L.L., Sudholt, D. and Yao, X., 2014, "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.
14. Mohammad ,S.Esam ,2019, "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.
15. P.W. Tsai, J. S. Pan, B. Y. Liao, M. J. Tsai, and V. Istanda, 2011, "Bat algorithm inspired algorithm for solving numerical optimization problems," *Appl.Mech. Mater.*, vols. 148\_149, pp. 134\_137.
16. Pressman, R.S., 2010, "Software engineering: a practitioner's approach". McGraw Hill.
17. S.-S. Guo, J.-S. Wang, and X.-X. Ma, 2019, "Improved bat algorithm based on multipopulation strategy of island model for solving global function optimization problem", *Comput. Intell. Neurosci.*, vol. 2019, pp. 1-12 ,Aug.
18. Vitekar, A.K.N., Dhanawe, S.A. and Hanchate, D.B., 2013, "Review of Solving Software Project Scheduling Problem with Ant Colony Optimization". *Journal International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 2, No. 4, pp.1177-1182.

19. Wu, X., Consoli, P., Minku, L., Ochoa, G. and Yao, X., 2016, September. "An evolutionary hyper-heuristic for the software project scheduling problem". In International Conference on Parallel Problem Solving from Nature Springer, Cham, pp.37-47.
20. X.-S. Yang,2010, "A new metaheuristic bat-inspired algorithm", in Nature Inspired Cooperative Strategies for Optimization. Berlin ,Germany: Springer, pp. 6574.
21. Y. Wang, P. Wang, J. Zhang, Z. Cui, X. Cai, W. Zhang, and J. Che,2019,"A novel bat algorithm with multiple strategies coupling for numerical optimization", Mathematics, vol. 7, no. 2, p. 135.