

Adaptive Mechanism For Enhanced Performance Of Shark Smell Optimization

N.A Kamarzaman, S.I Sulaiman, and I.R Ibrahim

Abstract— Various meta-heuristic approaches have been developed to find the optimal solution to optimization problems. However, different approach takes different amount of time and efficiency to achieve the optimal solution. Determination of high performance and lower computing time with simple algorithm is therefore continuously established. Shark Smell Optimization (SSO) algorithm has been proven to have high efficiency in many optimization applications. However, like the other swarm intelligence, SSO algorithm also has possibility to get trapped in local maxima or premature convergence. Thus, a new adaptive shark smell optimization (ASSO) is proposed to improve the convergence efficiency of standard SSO algorithm. An overview and performance comparison of six well-known meta-heuristic optimization algorithm is also presented in this paper. In order to verify the effectiveness of this newly developed method, the algorithm was tested on common benchmark functions used in the literature. Numerical results indicate that the ASSO algorithm strategy outperforms the basic SSO algorithm, Genetic Algorithm (GA), Particle Swarm Intelligence (PSO), Firefly Algorithm (FA), Artificial Bee Colony (ABC) and Teaching Learning Based Optimization (TLBO) in term of reaching for global solution.

Index Terms— meta-heuristic algorithm, shark smell optimization, adaptation mechanism, benchmark function, global optimization

I. INTRODUCTION

OPTIMIZATION is performed to achieve the maximization or minimization of a non-linear problem. In many applications nowadays, achieving global best optimization is significantly important. For many decades, a lot of optimization methods have been developed to solve global optimization.

Optimization can be divided into two categories; deterministic and stochastic. The deterministic methods require less iteration and its calculation is simple. However, they apply local information for the optimization and are unable to deliver satisfactory results under uncertainty conditions or challenging problems [1]–[3]. This is because the methods are inefficient when solving large scale optimization components besides being inflexible to adapt the solution to a given problem. Other than that, the optimization requires some assumption that might not be easily validated in many situations [1]. Hence, they converge only at the local minimum point of the objective

This manuscript is submitted on 6th October 2020 and accepted on 8th December 2020.

N.A Kamarzaman and S.I Sulaiman are with the Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor while I.R Ibrahim is with Faculty of Electrical Engineering, Universiti Teknologi MARA, 13500 Permatang Pauh, Pulau Pinang (e-mail: shahril_irwan2004@yahoo.com)

function. While, stochastic methods possess some randomness strategy during the optimization process [1]. Even though these methods are more complex, the algorithms able to escape from trapping in local optima while searching in a large-scale region. Meta-heuristic algorithms which are inspired by the nature of biological process such as genetic algorithm or the simulation of the collective motion of animals are usually stochastic in nature. Such kind of methods are capable of overcoming problems arise from classical method.

Nevertheless, according to no-free-lunch (NFL) theorem, there is no universal algorithm that can outperform all problems. Therefore, the development of a meta-heuristic optimization algorithm with high convergence and lower computation time is often a constant challenge. In recent years, many new advancements in optimization methods were proposed by the researchers with different perspectives and varying degree of success.

This paper presents performance comparison of selected well-known nature-inspired meta-heuristic algorithm; genetic algorithm (GA), particle swarm optimization (PSO), firefly algorithm (FA), artificial bee colony (ABC), teaching-learning based optimization (TLBO) and shark smell optimization (SSO). A new adaption mechanism to enhance the quality of standard SSO algorithm is proposed in this paper. A short overview of the stated meta-heuristic methods, methodology for standard SSO and ASSO algorithm and benchmark functions used in this paper are presented in section II. The numerical results and performance comparison are discussed in Section III and the last section presents the conclusion of this paper.

II. MATERIALS AND METHOD

In this section, the development of Adaptive Shark Smell Optimization (ASSO) is presented. Firstly, an overview of meta-heuristic optimization is discussed. Then, the formulation of adaptive mechanism in SSO is explained. Lastly, the benchmark functions used for comparing the performance of the proposed ASSO with selected optimization techniques are defined and elaborated.

A. Overview of meta-heuristic optimization

The literature recorded various meta-heuristic optimization methods developed. Meta-heuristic methods offer a better trade-off between exploration and exploitation in the optimization. The results generally deliver a promising high-quality solution to solve real-life problems. This section discusses selected six most popular metaheuristic algorithms; genetic algorithm, particle swarm optimization, firefly algorithm, artificial bee colony, teaching-learning based

algorithm and shark smell optimization.

1) Genetic Algorithm

Genetic algorithm (GA) is an evolutionary-based algorithm inspired by Charles Darwin's theory of natural genetics formation and selection. Genetic operators which are selection, crossover and mutation are the essential parts in problem-solving strategy applied in GA.

The fittest of the solution is selected by the selection operator based on the objective function values. The selection operator controls the search behaviour of the operator and retains only the best solution to generate a new population while the bad solution will be eliminated. The individuals in a new population are obtained by the crossover and mutation operator [4]–[6]. Crossover which is also known as recombination combines the genetic information of two parents or solutions to provide a newly generated solution or offspring. This is an important stage in genetic algorithm. In crossover, the solution will converge near to the optimal solution. Mutation operator refers to the phase of genetic algorithm reproduction. Mutation operator randomly changes the current solution to provide a wide diversity population. Thus, avoiding the solution from being trapped in the local optimum.

Genetic algorithm has been applied in vast applications due to its advantage that able to deal with complex problem and various types of optimization. Genetic algorithm has been developed and modified in many variants to be applied in a wide range of optimization problems [7]. Besides that, there are many modern developed algorithms nowadays which are either inspired by evolutionary algorithm of GA or have some strong similarities with it. However, genetic operators of GA parameters such as rate of mutation and crossover and new population selection need to be properly chosen. Inappropriate choice will make the algorithm difficult to converge and lead to slow convergence or not producing a valid result [8].

2) Particle Swarm Optimization

Particle swarm optimization (PSO) was developed by Eberhart and Kennedy inspiring from swarm behaviour in nature such as fish and bird schooling. PSO is the first nature-inspired meta-heuristic algorithm developed based on swarm intelligence. Unlike GA, PSO algorithm is much simpler, flexible and easy to be implemented in the area of optimization. Rather than applying genetic operators, PSO implements global communication among the swarm particle in the population based on random exploitation and exploration. Thus, no decoding and encoding parameters into binary string is required [9].

PSO algorithm is based on movement of particles until global optimum objective function is achieved. Each of the particles will move randomly to find a location that is better than the previous location. Particles position and velocity will be updated until their new current best location or local optimum is determined. The searching process continues among the particles in the population until the global best current location is achieved based on the objective function or user-defined maximum number of iterations.

PSO has been widely used in many areas such as optimization, computer intelligence and design application [10]. Even though the algorithm is simpler and easy to be applied, PSO algorithm easy to fall in local optima besides having a slow convergence [1], [11], [12]. These occurred because PSO is sensitive to inertia weight and poor tuning of this parameter will cause PSO to converge prematurely. Hence, there are many variants of PSO have been developed nowadays involving modification, adaptation mechanism or hybridization with other computer intelligence-based algorithm [1],[13].

3) Firefly Algorithm

Firefly algorithm (FA) is another meta-heuristic algorithm that is based on swarm intelligence. FA was developed by Xin-She Yang, by the basis of flashing pattern and behaviour of fireflies [14]. The flashing pattern of fireflies is unique among the species and is used in attracting partner or potential prey, communication, or protecting themselves from the predators.

The base of firefly algorithm is established by three idealized assumptions [15]. It is assumed that the sex category is ignored. Firefly can be attracted to each other without considering their sex. Then, the level of brightness determines the attraction of fireflies. The farther the distance, the brightness will decrease. Any less bright firefly will move towards the brighter ones or if there is no brighter flashing light, the firefly will move randomly, and the objective function is formulated based on the brightness of firefly [1].

Inspired by the firefly movement in nature, firefly algorithm is performed by three significant parameters; attractiveness, random movement and light absorption. The attractiveness depends on the level of brightness. Firefly algorithm can be automatically divided into a few other groups in the population based on the fitness flashlight intensity of fireflies. The individual firefly is forced to move either systematically or randomly towards the brighter flashing one until the population converges to the brightest state to find the global optimum. Considering light absorption in the media, the attractiveness varied according to the level of absorption [1].

The auto-subdivision of FA allows the algorithm to find local optima simultaneously. Hence, this algorithm is good and efficient in determining the local optimum value [16]. However, the drawbacks of this algorithm are slow convergence speed and high possibility to be trapped in local optima while having more control variables [17]. FA has been developed in many variants mainly to improve its performance and increase convergence speed. For instance, FA parameters can be tuned and modified to control the random movement of FA to increase the convergence speed.

4) Artificial Bee Colony

Artificial bee colony (ABC) algorithm is modelled from the natural behaviour of real honeybees in food foraging. This algorithm was proposed in the area of optimization by Karaboga in 2005 [18]. There are three groups of bees with different purposes involved in the algorithm.

The exploitation for the food source starts with employed bees. The employed bees carry information about the specific

food source and this will be shared with the onlooker bees. The information shared includes the distance, the direction and the profitability of the food source. Then the onlooker bees will evaluate the quality of the food source based on the employed bee waggle dance. The other phase is the scout bees which are the abandoned bees that look for the other new food source randomly [19]–[21].

In the exploration and exploitation of ABC algorithm, a food source's location presents a potential solution to the optimization problem. The amount of food source's nectar correlates to the fitness value of the corresponding solution [22]. It is assumed that one employed bee is associated to one food source. The number of employed bees therefore represents the number of food sources. When the artificial bees find a new food source with higher nectar value than the previous, ABC is considered moving to a better solution and the poorer solution in previous state will be ignored.

Self-organizing and collective intelligence makes the ABC algorithm robust, and therefore avoid the algorithm traps in the local optima [23],[24].

5) Teaching learning-based optimization

Teaching learning-based optimization algorithm was developed by Rao et al. in 2012. It is modelled by the principal of teaching and learning, in which teachers influence the learner's performance in a class [25], [26]. Unlike the other optimization algorithm, TLBO algorithm does not require some specified algorithm-controlling parameters such as mutation and crossover probability in GA or inertia weight in PSO.

Formulation of TLBO requires two fundamental phases. The first phase is the teacher phase. The teacher phase will be considered as the best solution or optimum objective function based on the nature of teaching and learning. In the actual condition, the teacher is mostly more knowledgeable and highly learned than the students. The second phase is presented by the learner's phase. The learning process occurred when there is an interaction between the students, and this is considered as the solution to the problem.

TLBO only requires the size of the population and the design variables as algorithm control parameters. The population size of the algorithm is depicted by the number of learners in a class and different subjects to be taught to the learners are considered as variable design parameters of the optimization problem while the fitness function of the problem is valued by the learner's performance. Final grades or marks will be considered as the output of the algorithm [25].

In the teacher's phase, the algorithm attempts to improve the learners by moving their position towards the teacher position. The improvement of the learners will be influenced by the difference between the knowledge of teachers and quality of the overall students. While in the learner's phase, the improvement is by peer learning between the learners. Learners in the current position will move to the smarter learner and move away if otherwise. A new learner will be accepted into the population if it can perform better [27]. The iteration is terminated when the maximum number of generations is reached.

Since TLBO requires less evaluation, optimization results

from the literature show that the convergence speed of TLBO is faster compared to the other optimization method. However, since the result is dependent on the teacher and interaction with other learners, it has the probability of not achieving the optimum result.

6) Shark Smell Optimization

Shark smell optimization (SSO) is an algorithm that was proposed by Abedinia et al. in 2014, which was inspired by the natural behaviour of shark in hunting for the prey in the sea water. As an apex predator, a shark has a great sense of smell that can detect a single drop of blood in the large ocean. The acute nostrils able to sharply detect the source of odour. Thus, enable shark to turn to the direction of the potential prey. Lateral lines that run down each side of the shark's body allow the detection of any vibrations or pulses in the water produced by the prey [28]. This great smell sense to capture prey becomes the reason for shark survival in the ocean.

SSO has simple calculation that requires only updated velocity and position of shark, similar to PSO. The movement of shark is strongly dependent on the odour concentration of the injured prey. This means that shark will move towards a higher concentration of the odour in a forward and rotational movement. The introduction of local search made capable a more accurate and efficient searching process. Besides that, the algorithm is easy to be implemented to a program. Therefore, SSO algorithm has incomparable performance than other meta heuristic algorithm. Until now, SSO algorithm has been applied in many areas of optimization such as in transportation, power system, medical, control system and energy system [29]–[31]. Besides that, it has also been modified in few versions to improve its capability [32]–[34].

Characteristic and standard parameters of the mentioned meta-heuristic algorithm in this paper are given in Table 1. The advantages of an algorithm over the other are also stated in the Table 1.

B. Shark Smell Optimization Algorithm

1) Standard SSO algorithm

Shark has not only an accurate smell sense but also known for its foraging movement of forward and rotation that help it becomes a prevailing hunter in the ocean [35]. The initialization of standard SSO algorithm starts with the shark searching process when detecting odour particles from injured prey within a randomly generated initial position of the shark. Each injured prey is considered to have one blood source. The standard SSO algorithm is formulated based on [28]. The current position can be expressed as:

$$X_i^l = [x_{i1}^1, x_{i1}^2, \dots, x_{i1}^{NP}] \quad (1)$$

Shark moves to the location of an injured prey when detecting blood odour, with a specific velocity defined by:

$$V_i^l = [v_{i1}^1, v_{i1,2}^1, \dots, v_{i1}^{ND}] \quad (2)$$

The shark's velocity increases when detecting high blood odour. Hence in optimization, the odour concentration can be considered as a gradient of the objective function. The velocity

is varied according this gradient of objective function.

$$V_i^k = \eta_k \cdot R1 \cdot \frac{\delta(OF)}{\delta x_j} | x_i^k, \quad j = 1, \dots, ND, i = 1, \dots, NP, k = 1, \dots, kmax \quad (3)$$

Shark accelerates at a specific velocity subjected to an inertial limitation.

$$V_i^k = \eta_k \cdot R1 \cdot \nabla(OF) | x_i^k + \alpha_k \cdot R2 \cdot v_{i,j}^{k-1}, \quad (4)$$

α_k is momentum rate and the value is in interval [0,1]. R2 is a random value for momentum term. The current velocity of sharks strongly depends on the previous velocity as in (4). R1 and R2 in random number in range of [0,1]. Meanwhile, the acceleration of sharks is limited by a velocity limiter by following equation:

$$|v_i^k| = \min [| \eta_k \cdot R1 \cdot \nabla(OF) | x_i^k + \alpha_k \cdot R2 \cdot v_{i,j}^{k-1} |, | \beta_k \cdot v_{i,j}^{k-1} |] \quad (5)$$

β_k is velocity limiter. The updated position of shark in forwarding motion is given by:

$$Y_i^{k+1} = X_i^k + V_i^k \cdot \Delta_{tk}, \quad (6)$$

where Y_i^{k+1} is the new shark's position and Δ_{tk} is time interval. Other than that, sharks also implement a rotational movement for a local search in each stage when searching for a better candidate solution. The equation is given by:

$$Z_i^{k+1,m} = Y_i^{k+1} + R_3 \cdot Y_i^{k+1}, \quad m = 1, \dots, M \quad i = 1, \dots, NP \quad k = 1, \dots, kmax \quad (7)$$

where R_3 is a random number in the range [-1,1] and M is the total points in local search in rotational movement and m is the number of each rotation level. When connecting M points surround local search, a close contour like rotational movement of shark can be obtained. The next position of sharks is selected based on the best position obtained between forward movement, Y_i^{k+1} and rotational movement, $Z_i^{k+1,m}$ until reach k_{max} .

Pseudo-code of SSO algorithm

Begin

Step 1. Initialization

Set parameters $NP, k_{max}, \eta_k, \alpha_k, \beta_k, (k = 1, 2 \dots k_{max})$

Generate individual's initial population

Generate each decision variable randomly

Initialize the stage counter $k = 1$

For $k = 1 : k_{max}$

Step 2.

Forward movement

Calculate the updated velocity vector.

Obtain new position of shark based on the forward movement.

Rotational movement

Obtain position of shark based on rotational movement, $Z_i^{k+1,m} (m = 1, \dots, M)$

Select next the best position of shark based on the two

movements, rotational and forward movements.

End for k

Set $k = k + 1$

Step 3. Best position

Select the best position of shark based on the optimum objective function in the last step of iteration.

End

2) Adaptive Shark Smell Optimization

Even though the basic SSO algorithm has been utilized in various optimization problems, there is still room for improvement. Some adaption mechanism is required to improve its searching ability and convergence behaviour. In the exploitation and exploration of SSO algorithm, parameters for gradient term and inertia term are determined randomly during the initialization of algorithm. Gradient term is used for a random probability searching pattern and momentum term is used to improve searching diversity randomly. Meanwhile, parameter for velocity limiter is set based on the value of maximum and minimum velocity of shark [28]. This parameter is fixed and does not change throughout the iteration.

These parameter's conditions issue possibility for the solution to be trapped in local maxima. Therefore, this study proposes ASSO algorithm for gradient term, momentum term and velocity limiter parameter to improve SSO searching ability for global solution. In order to achieve that, some adaption will be included in these parameters.

3) Adaptive velocity limiter term

The velocity of shark hunts the injured prey is bounded by a range of certain velocity values. This is given by:

$$V_i^k = \begin{cases} V_{max}, & \text{for } V_i^k > V_{max} \\ V_{min}, & \text{for } V_i^k \leq V_{min} \end{cases}$$

This velocity control or velocity limiter parameter is based on the ratio of V_{max} to V_{min} which has been added into the equation (4). According to [28], the minimum shark's velocity is 20km/h while its maximum velocity is 80km/h. Thus V_{max} is set 80 while V_{min} is set 20. However, in a certain test function, the algorithm tends to premature converge or failed to converge to the optimum value.

Thus, this study proposed an adaptation mechanism to be added to the formulation of SSO algorithm. The proposed ASSO algorithm is empirically determined. Linear decreasing velocity limiter term is proposed in the velocity update equation so that the algorithm will start with a large step at the beginning of iteration and a smaller step towards the end of iteration. In this way, the algorithm will have high exploration capability. The velocity limiter is calculated in the following formula:

$$\beta_k = \left(\frac{V_{max}/V_{min}}{\text{No of population}} \right) * \omega \beta_k \quad (8)$$

$$\omega \beta_k = \omega_{start} - \frac{\omega_{start} - \omega_{end}}{Maxiter} * ithiter \quad (9)$$

Where ω_{start} and ω_{end} stand for the starting and ending values for the velocity limiter coefficient of SSO algorithm which is set to 1 and 0 respectively, $iter_{max}$ and $iter$ represent the total number of iteration and current iteration value. The $\omega\beta_k$ and the maximum number of iterations is inversely proportional. When the iteration increases, the value of $\omega\beta_k$ decreases, from 1 to 0. Hence, the algorithm will be able to operate with high efficiency in a shorter time.

4) Adaptive gradient and inertia term

On the other hand, to improve the efficiency of the algorithm, linear decreasing η_k term and linear increasing a term have also been added into the formulation of SSO algorithm.

This is given by:

$$\omega\eta_k = \omega_{start} - \frac{\omega_{start} - \omega_{end}}{MaxIter} * ith_{iter} \quad (10)$$

$$\omega\alpha_k = \omega_{end} - \frac{\omega_{end} - \omega_{start}}{iter_{max}} * iter \quad (11)$$

Where $\omega\eta_k$ is the adaptation mechanism for gradient term and $\omega\alpha_k$ is the adaptation mechanism for the inertia term. The value of $\omega\alpha_k$ is directly proportional to the total number of iterations where $\omega\alpha_k$ is increasing towards the maximum number of iterations and $\omega\eta_k$ is vice versa. Thus, $\omega\alpha_k$ is varied from 0 to 1 while $\omega\eta_k$ varies from 1 to 0 through out the iteration.

The velocity vector in equation (4) with the added adaption mechanism given as below.

$$|v_i^k| = \min [| \omega\eta_k \cdot RI \cdot \nabla(OF) | x_i^k + \omega\alpha_k \cdot R2 \cdot v_{i,j}^{k-1}, | \omega\beta_k \cdot v_{i,j}^{k-1} |] \quad (12)$$

Pseudocode adaptive Shark Smell Optimization algorithm

Begin

Step 1. Initialization

Set parameters of NP and k_{max}
Generate individual's initial population
Generate each decision variables randomly
Initialize stage counter, $k = 1$

For $k = 1: k_{max}$

Step 2.

Forward movement

Determine parameter coefficient in the velocity vector
Calculate $\omega\beta_k$ by equation (9)
Calculate $\omega\eta_k$ by equation (10)
Calculate $\omega\alpha_k$ by equation (11)
Calculate the updated velocity vector
Update the new position of shark based on the forward movement

Rotational movement

Obtain new position of shark based on rotational movement
Select the new position of shark from the updated position of forward and rotational movement
End for k

Set $k = k + 1$ until k_{max}

Step 3. Best position

Select the best position of shark based on the optimum objective function in the last step of iteration.

End

TABLE 1
CHARACTERISTICS COMPARISON OF META-HEURISTIC OPTIMIZATION METHOD

No	Method	Standard Parameter	Biological Inspiration	Advantage
1	GA	Population size Selection Crossover Mutation,	Inspired by the theory of natural evolution	Vast application
2	PSO	No. of population Learning rate Inertia weight	Social behaviour of bird flocking or fish schooling	Simple algorithm
3	FA	Light intensity variation Attractiveness of firefly	Process of bioluminescence of firefly	Efficient determine local maxima
4	ABC	No. of population Employed and unemployed foraging bees Food sources	Natural foraging of bee honey	Less control parameters
5	TLBO	Population size No. of generations	Model from teaching & learning principal in classroom	Fast convergence
6	SSO	No. of population Gradient coefficient Momentum rate Velocity limiter	Inspired from shark hunt of the injured prey in the ocean	Simple algorithm

C. Benchmark optimization functions

In order to fairly compare performance of the optimization algorithms, a set of special designed test problems is required. Each of the algorithms is run multiple times for each problem to accurately determine the optimum objective function value. This process is called benchmarking. Benchmark test function is significant to test the effectiveness of a newly developed algorithm in which the characteristic of the function is still unknown.

In this section, the capability and effectiveness of ASSO algorithm is assessed by solving multiple benchmark function. Six benchmark functions that are generally used in the literature are considered in this paper to compare the performance of ASSO with other selected meta-heuristic algorithms.

The statistical data are including the maximum index value, minimum index value, mean index value, standard deviation value and convergence speed. Convergence speed is assessed by the basis of an algorithm that has reached the minimum global solution with the fastest elapsed time.

The algorithm is evaluated based on the following mechanism [36]:

1. The algorithms that have been able to obtain the

minimum function value are chosen. Based on Table 2, the global solution for every benchmark function is the minimum value given by F_{\min} .

2. The algorithm with the shortest convergence speed is chosen.
3. Subsequently, the evaluation of algorithm robustness and efficiency is established by the low mean value.

Generally, test optimization using benchmark function can be divided into two groups; unimodal and multimodal. When plotted in 3D, the shape of unimodal benchmark functions has one optimum value compared to multimodal that has one global optimum in many local optima.

The $f_1 - f_3$ are unimodal benchmark functions that have only one global best value. This kind of test function is normally used to assess local exploitation capability [37]. Hence, the rate of convergence is more significant to be considered compared to the quality of the final results. Hence, convergence speed will be observed in unimodal optimization. On the other hand, multimodal benchmark functions, $f_4 - f_6$, have multiple global best values. The multimodal function is suitable to be used to investigate the quality of optimization method by assessing the exploration capability to escape trapping in the local maxima [14], [37].

Maximum, minimum and mean index values after final iteration were considered to assess the quality of optimization method. From Table 2, the minimum value for all functions is zero. This means that each function needs to achieve zero value for global optimization. The mean index indicates the accuracy and searchability.

III. RESULTS AND DISCUSSION

In this section, the effectiveness of the proposed adaptive method has been checked with six benchmark test functions. The performance of the ASSO algorithm is compared with the other six selected meta-heuristic methods.

The setting of the standard parameters GA, PSO, FA, ABC, TLBO, standard and ASSO algorithm, are as follows:

1. Number of population (candidate) = 30;
2. Maximum number of iterations (generations) = 100.

The optimization results for each benchmark function are presented in Table 3 to Table 8.

In all benchmark test functions, each of the algorithms was run 30 times and the results were taken at the 30th simulation run. The numerical results consist of minimum, maximum, mean and standard deviation of the objective function as well as the execution period. Standard deviation is represent as std dev in the table.

UNIMODAL AND MULTIMODAL BENCHMARK FUNCTION				
Name	Function	D	S	F_{\min}
Sphere	$f_1(X) = \sum_{i=1}^D X_i^2$	30	[100,100] ^D	0
Zakharov	$f_2(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4$	30	[-5,10]	0
Matyas	$f_3(x, y) = 0.26(x^2 + y^2) - 0.48xy$	30	[-10,10]	0
Rastrigin	$f_4(X) = \sum_{i=1}^D [X_i^2 - 10\cos(2\pi X_i) + 10]$	30	[-5.12, 5.12] ^D	0
Ackley	$f_5(X) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n X_i^2}\right)$	30	[-32,32] ^D	0
Griewank	$f_6(X) = \frac{1}{4000}\sum_{i=1}^n X_i^2 - \prod_{i=1}^n \cos\left(\frac{X_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0

TABLE 3
OBTAINED RESULT FOR SPHERE BENCHMARK FUNCTION

Result	GA	PSO	FA	ABC	TLBO	SSO	A SSO
Time (s)	0.2	4.34	1.18	3.9	0.32	0.47	0.62
MIN	25	0.89	2.4	1.16	2.04	1.8 × 10 ⁻²¹⁸	0
MAX	38.2	68.1	51.1	73.91	2.684	2.44 × 10 ⁻⁴	1.27 × 10 ⁻⁴
Mean	25.97	35.6	1.11	2.415	0.032	2.53 × 10 ⁻⁶	1.28 × 10 ⁻⁶
Std dev	2.007	22.5	6.02	9.959	0.268	2.43 × 10 ⁻⁵	1.26 × 10 ⁻⁵

TABLE 4
OBTAINED RESULT FOR ZAKHAROV BENCHMARK FUNCTION

Result	GA	PSO	ABC	TLBO	FA	SSO	A SSO
Time (s)	0.1	4.69	4.35	0.21	1.3	0.636	0.7
MIN	3.39 × 10 ⁻⁹	2.86 × 10 ⁻¹⁰	1.48 × 10 ⁻¹⁴	1.49 × 10 ⁻³⁷	1.56 × 10 ⁻⁸	0	0
MAX	3.5 × 10 ⁻²	1.689	4472.72	5	0.405	4.87 × 10 ⁻⁴	1.94 × 10 ⁻⁵
Mean	1.44 × 10 ⁻³	4.01 × 10 ⁻²	47.135	0.111	4 × 10 ⁻³	5.13 × 10 ⁻⁶	1.94 × 10 ⁻⁷
Std dev	6.59 × 10 ⁻³	0.212	445.109	0.624	4.03 × 10 ⁻²	4.98 × 10 ⁻⁵	1.94 × 10 ⁻⁶

TABLE 2

TABLE 5

OBTAINED RESULT FOR MATYAS BENCHMARK FUNCTION

Result	GA	PSO	FA	ABC	TLBO	SSO	A SSO
Time (s)	0.18	4.88	1.24	2.16	0.21	0.478	0.635
MIN	1.41 $\times 10^{-13}$	2 $\times 10^{-11}$	4.49 $\times 10^{-9}$	2.568 $\times 10^{-3}$	6.25 $\times 10^{-28}$	3.1 $\times 10^{-27}$	0
MAX	1.27 $\times 10^{-2}$	7.87 $\times 10^{-3}$	3.343 $\times 10^{-3}$	11.56	1.069	1.51 $\times 10^{-6}$	1.91 $\times 10^{-6}$
Mean	3.59 $\times 10^{-4}$	1.24 $\times 10^{-3}$	3.88 $\times 10^{-5}$	0.752	1.4 $\times 10^{-2}$	1.52 $\times 10^{-8}$	1.92 $\times 10^{-8}$
Std dev	1.83 $\times 10^{-3}$	2.53 $\times 10^{-3}$	3.35 $\times 10^{-4}$	1.566	0.107	1.51 $\times 10^{-7}$	1.9 $\times 10^{-7}$

TABLE 6

OBTAINED RESULT FOR RASTRIGIN BENCHMARK FUNCTION

Result	GA	PSO	FA	ABC	TLBO	SSO	A SSO
Time (s)	2.6 $\times 10^{-1}$	6.18	1.31	1.03	0.2	0.453	0.4756
MIN	0	5.1 $\times 10^{-5}$	4.92 $\times 10^{-6}$	1.42 $\times 10^{-14}$	0	0	0
MAX	7.473	5.994	0.198	100.5	9.39	7.25 $\times 10^{-2}$	0.132
Mean	0.363	1.637	4.21 $\times 10^{-3}$	3.559	0.6	4.52 $\times 10^{-2}$	1.344 $\times 10^{-3}$
Std dev	1.216	1.885	2.774	12.82	2.05	3.41 $\times 10^{-2}$	1.314 $\times 10^{-3}$

TABLE 7

OBTAINED RESULT FOR GRIEWANK BENCHMARK FUNCTION

Result	GA	PSO	FA	ABC	TLBO	SSO	A SSO
Time (s)	0.2	4.67	1.33	5.06	0.21	0.358	0.475
MIN	1.47 $\times 10^{-2}$	0.185	0.442	1.22 $\times 10^{-6}$	07.4 $\times 10^{-3}$	0	0
MAX	9.62 $\times 10^{-2}$	0.973	73.66	0.311	4.78 $\times 10^{-2}$	6.25 $\times 10^{-4}$	4.29 $\times 10^{-5}$
Mean	2.31 $\times 10^{-2}$	0.667	2.218	0.011	8.46 $\times 10^{-3}$	7.36 $\times 10^{-6}$	4.31 $\times 10^{-7}$
Std dev	1.95 $\times 10^{-2}$	0.209	8.297	0.036	5.65 $\times 10^{-3}$	6.3 $\times 10^{-5}$	4.27 $\times 10^{-6}$

TABLE 8

OBTAINED RESULT FOR ACKLEY BENCHMARK FUNCTION

Result (s)	GA	PSO	FA	ABC	TLBO	SSO	A SSO
Time	0.17	4.48	1.28	2.91	0.32	0.503	0.568
MIN	1.25 $\times 10^{-3}$	6.96 $\times 10^{-6}$	2.6 $\times 10^{-4}$	1.01 $\times 10^{-5}$	8.88 $\times 10^{-16}$	8.88 $\times 10^{-16}$	8.88 $\times 10^{-16}$
MAX	4.189	2.891	0.223	19.97	4.598	1.81 $\times 10^{-2}$	2.71 $\times 10^{-2}$
Mean	0.167	0.445	4.46 $\times 10^{-4}$	7.814	0.124	8.55 $\times 10^{-3}$	3.12 $\times 10^{-4}$
Std dev	0.659	0.818	2.21 $\times 10^{-2}$	8.941	0.603	6.35 $\times 10^{-3}$	2.73 $\times 10^{-3}$

As from Table 3 to Table 8, considering the minimum objective function value given by MIN index, the proposed ASSO algorithm able to achieve minimum value of zero at the end of iteration for all benchmark functions except in Ackley Function. In Ackley Function, ASSO, standard SSO and TLBO algorithm achieved the minimum value at 8.88×10^{-16} compared to GA, PSO, FA and ABC. The basic SSO algorithm able to achieve zero value for the minimum objective function

in Rastrigin, Griewank and Zakharov test function. Meanwhile, GA and TLBO able to achieve zero objection function in Rastrigin Function.

Regarding the mean value, lower value of mean index shows that the optimization methods has higher accuracy and good searching ability. From the table, ASSO algorithm able achieve the lowest MEAN index value for all test functions. This shows that the proposed adaptive algorithm is more accurate in finding global optimization compared to the other meta- heuristic methods.

In terms of time performance, it is established using TIC and TOC which is the starting and end time of the computer central processing unit (CPU). From all the numerical tables, the basic SSO algorithm and the ASSO algorithm solved all the benchmark functions in less than 1 second.

The experimental to verify the performance of the proposed algorithm in exploration and exploitation is extended with varying number of population in the algorithm. For 100 iterations, number of population is varied from $N = 50$, then $N = 100$ to $N = 500$ for a unimodal and multimodal benchmark function. Table 9 and 10 and Fig.1 and Fig.2 show mean index values for Matyas unimodal and Griewank multimodal benchmark function. As can be seen from the table and figures, ASSO algorithm has the lowest mean index value compare to the other meta-heuristic optimization method. Therefore, ASSO algorithm gives more satisfactory balancing between exploitation and exploration.

TABLE 9

MATYAS FUNCTION MEAN INDEX VALUE FOR VARYING NO OF POPULATION

No Pop	GA	PSO	FA	ABC	TLBO	SSO	ASSO
50	1.99 $\times 10^{-3}$	1.38 $\times 10^{-3}$	5.36 $\times 10^{-6}$	0.283	6.82 $\times 10^{-3}$	3.98 $\times 10^{-8}$	5.22 $\times 10^{-8}$
100	1.84 $\times 10^{-4}$	1.89 $\times 10^{-3}$	9.83 $\times 10^{-8}$	0.089	3.08 $\times 10^{-4}$	1.85 $\times 10^{-10}$	3.12 $\times 10^{-9}$
200	1.08 $\times 10^{-4}$	1.57 $\times 10^{-4}$	1.79 $\times 10^{-6}$	0.156	3.63 $\times 10^{-3}$	5.09 $\times 10^{-8}$	8.92 $\times 10^{-10}$
300	2.28 $\times 10^{-4}$	8.2 $\times 10^{-5}$	7.75 $\times 10^{-7}$	0.032	3.15 $\times 10^{-4}$	7.66 $\times 10^{-10}$	1.28 $\times 10^{-11}$
400	4.96 $\times 10^{-5}$	8.79 $\times 10^{-5}$	1.05 $\times 10^{-7}$	0.086	9.18 $\times 10^{-4}$	7.62 $\times 10^{-10}$	1.4 $\times 10^{-10}$
500	1.8 $\times 10^{-5}$	6.74 $\times 10^{-5}$	1.51 $\times 10^{-7}$	0.073	3.4 $\times 10^{-5}$	1.06 $\times 10^{-11}$	1.6 $\times 10^{-14}$

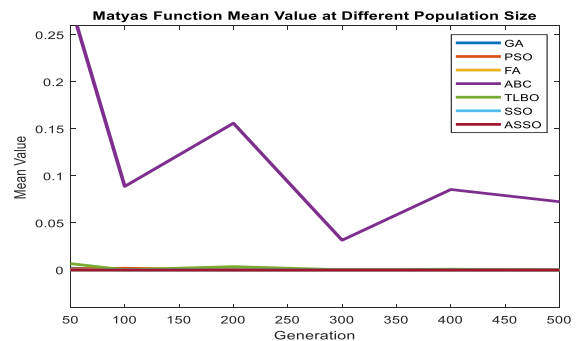


Fig. 1. Mean value for Matyas multimodal benchmark function varying no of population

TABLE 10

GRIEWANK FUNCTION MEAN INDEX VALUE FOR VARYING NO OF POPULATION

No Pop	GA	PSO	FA	ABC	TLBO	SSO	ASSO
50	0.024	0.740	0.0421	9.75×10 ⁻³	3.24×10 ⁻³	2.35×10 ⁻⁶	4.37×10 ⁻⁵
10	0.040	0.622	1.78×10 ⁻²	4.89×10 ⁻³	7.25×10 ⁻³	1.05×10 ⁻⁵	4.55×10 ⁻⁶
20	0.021	0.649	1.19×10 ⁻²	4.89×10 ⁻³	4.03×10 ⁻⁴	3.69×10 ⁻⁷	5.67×10 ⁻¹²
30	0.018	0.498	8.81×10 ⁻³	3.21×10 ⁻³	1.44×10 ⁻³	9.23×10 ⁻¹⁰	8.49×10 ⁻⁹
40	0.021	0.216	5.44×10 ⁻³	3.49×10 ⁻³	1.06×10 ⁻³	1.48×10 ⁻⁷	2.61×10 ⁻¹⁰
50	0.021	0.307	5.72×10 ⁻³	2.79×10 ⁻³	3.56×10 ⁻⁴	9.56×10 ⁻⁹	1.69×10 ⁻⁹
0	1	7	0 ⁻³	0 ⁻³	0 ⁻⁴	0 ⁻⁹	0 ⁻⁹

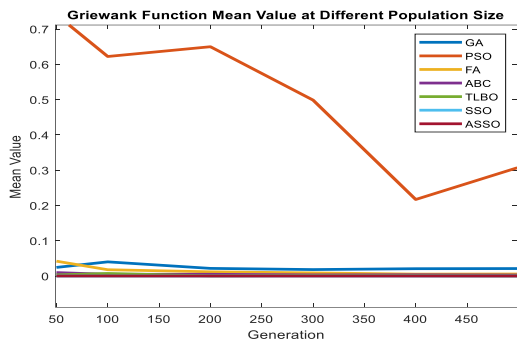


Fig. 2. Mean value for Griewank multimodal benchmark function varying no of population

From the quality of numerical results and time performance, the ASSO algorithm is very useful to improve the searching ability and efficiency of the SSO algorithm.

IV. CONCLUSION

The main contribution of this paper is to propose an adaptive mechanisms to enhance the performance of the standard SSO algorithm. The statistical mean index is used to indicate the stability of the algorithm in the exploitation and exploration. The results reveal that with the adaption mechanism, ASSO algorithm benefits in high exploration which ASSO algorithm achieve minimum mean index value in the numerical experiment. Other than that, ASSO algorithm able to achieve global optimization with $F_{min} = 0$ in most benchmark functions or the nearest optimal solution. The algorithm also succeeds in solving the problem in the fastest time compared to the other tested meta-heuristic algorithms such as PSO, ABC, TLBO, FA, GA and the standard SSO algorithm.

For the future works, the proposed ASSO algorithm will be implemented and tested in real application such as power system optimization to further verify the results as well improve the performance of the system.

REFERENCES

[1] Xin-She Yang, *Nature-Inspired Algorithms and Applied Optimization*. Springer International Publishing AG, 2018.
 [2] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Inf. Sci. (Ny)*, vol. 540, pp. 131–159, 2020.

[3] F. Zhu *et al.*, "Short-term stochastic optimization of a hydro-wind-photovoltaic hybrid system under multiple uncertainties," *Energy Convers. Manag.*, vol. 214, no. 1, p. 112902, 2020.
 [4] D. A. Elvira-Ortiz, A. Y. Jaen-Cuellar, D. Morinigo-Sotelo, L. Morales-Velazquez, R. A. Osornio-Rios, and R. de J. Romero-Troncoso, "Genetic algorithm methodology for the estimation of generated power and harmonic content in photovoltaic generation," *Appl. Sci.*, vol. 10, no. 2, 2020.
 [5] K. Sasidhar and B. J. Kumar, "Optimal sizing of PV-Wind Hybrid energy system using Genetic Algorithm (GA) and Particle swarm optimization (PSO)," *Int. J. Sci. Eng. Technol. Res.*, vol. 4, no. 2, pp. 354–358, 2015.
 [6] N. Ghorbani, A. Kasaeian, A. Toopshekan, L. Bahrami, and A. Maghami, "Optimizing a hybrid wind-PV-battery system using GA-PSO and MOPSO for reducing cost and increasing reliability," *Energy*, vol. 154, 2018.
 [7] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12363–12379, 2020.
 [8] X. S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," *J. Comput. Sci.*, no. xxxx, p. 101104, 2020.
 [9] Xin-She Yang, "Particle Swarm Optimization," in *Nature-Inspired Optimization Algorithms*, Elsevier Inc., 2014, pp. 99–110.
 [10] Y. J. Gong *et al.*, "Genetic Learning Particle Swarm Optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277–2290, 2016.
 [11] Z. Yong, Y. Li-juan, Z. Qian, and S. Xiao-yan, "Multi-objective optimization of building energy performance using a particle swarm optimizer with less control parameters," *J. Build. Eng.*, vol. 32, no. October 2019, p. 101505, 2020.
 [12] M. Isiet and M. Gadala, "Sensitivity analysis of control parameters in particle swarm optimization," *J. Comput. Sci.*, vol. 41, p. 101086, 2020.
 [13] G. Xu, X. Zhao, T. Wu, R. Li, and X. Li, "An Elitist Learning Particle Swarm Optimization with Scaling Mutation and Ring Topology," *IEEE Access*, vol. 6, no. April, pp. 78453–78470, 2018.
 [14] L. Zhang, L. Liu, X. S. Yang, and Y. Dai, "A novel hybrid firefly algorithm for global optimization," *PLoS One*, vol. 11, no. 9, pp. 1–17, 2016.
 [15] N. I. Abdul Aziz, S. I. Sulaiman, S. Shaari, I. Musirin, and K. Sopian, "Optimal sizing of stand-alone photovoltaic system by minimizing the loss of power supply probability," *Sol. Energy*, vol. 150, pp. 220–228, 2017.
 [16] X. S. Yang, S. Deb, T. Hanne, and X. He, "Attraction and diffusion in nature-inspired optimization algorithms," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1987–1994, 2019.
 [17] M. Gupta and D. Gupta, "A New Modified Firefly Algorithm," *Int. J. Recent Contrib. from Eng. Sci. IT*, vol. 4, no. 2, pp. 18–23, 2016.
 [18] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Appl. Soft Comput. J.*, vol. 8, no. 1, pp. 687–697, 2008.
 [19] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artif. Intell. Rev.*, vol. 52, pp. 2191–2233, 2019.
 [20] K. Hussain, M. N. Mohd Salleh, S. Cheng, Y. Shi, and R. Naseem, "Artificial bee colony algorithm: A component-wise analysis using diversity measurement," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 7, pp. 794–808, 2020.
 [21] S. Anuar, A. Selamat, and R. Sallehuddin, "A modified scout bee for artificial bee colony algorithm and its performance on optimization problems," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 28, no. 4, pp. 395–406, 2016.
 [22] Xin-She Yang, *Engineering Optimization - An Introduction with Metaheuristic Applications*. John Wiley & Sons, 2010.
 [23] A. Choubey and G. . Prajapati, "An Understanding of ABC Algorithm and Its Application," *Int. J. Curr. Eng. Sci. Res.*, vol. 2, no. 10, pp. 105–109, 2015.
 [24] W. Mao, H. Y. Lan, and H. R. Li, "A New Modified Artificial Bee Colony Algorithm with Exponential Function Adaptive Steps," *Comput. Intell. Neurosci.*, vol. 2016, no. i, 2016.
 [25] R. V. Rao and V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems," *Int. J. Ind. Eng. Comput.*, vol. 3, no. 4, pp. 535–560, 2012.
 [26] R. V. Rao and V. Patel, "Sharif University of Technology An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Sci. Iran.*, vol. 20, no. 3, pp.

- 710–720, 2013.
- [27] J. H. Cho, M. G. Chun, and W. P. Hong, “Structure optimization of stand-alone renewable power systems based on multi object function,” *Energies*, vol. 9, no. 8, pp. 1–19, 2016.
- [28] O. Abedinia, N. Amjadi, and A. Ghasemi, “A new metaheuristic algorithm based on shark smell optimization,” *Complexity*, vol. 21, no. 5, pp. 97–116, 2014.
- [29] B. T. Ahmed and O. Y. Abdulhameed, “Fingerprint Authentication using Shark Smell Optimization Algorithm,” *UHD J. Sci. Technol.*, vol. 4, no. 2, p. 28, 2020.
- [30] L. Wang, X. Wang, Z. Sheng, and S. Lu, “Multi-objective shark smell optimization algorithm using incorporated composite angle cosine for automatic train operation,” *Energies*, vol. 13, no. 3, 2020.
- [31] P. Singh and B. Khan, “Smart microgrid energy management using a novel artificial shark optimization,” *Complexity*, vol. 2017, 2017.
- [32] M. Ehteram, A. N. Ahmed, C. M. Fai, H. A. Afan, and A. El-Shafie, “Accuracy enhancement for zone mapping of a solar radiation forecasting based multi-objective model for better management of the generation of renewable energy,” *Energies*, vol. 12, no. 14, 2019.
- [33] M. Bagheri, A. Sultanbek, O. Abedinia, M. S. Naderi, M. S. Naderi, and N. Ghadimi, “Multi-objective Shark Smell Optimization for Solving the Reactive Power Dispatch Problem,” *Proc. - 2018 IEEE Int. Conf. Environ. Electr. Eng. 2018 IEEE Ind. Commer. Power Syst. Eur. EEEIC/I CPS Eur. 2018*, no. January 2019, pp. 1–6, 2018.
- [34] O. Abedinia and N. Amjadi, “Short-term wind power prediction based on Hybrid Neural Network and chaotic shark smell optimization,” *Int. J. Precis. Eng. Manuf. - Green Technol.*, vol. 2, no. 3, pp. 245–254, 2015.
- [35] M. Ahmadigorji and N. Amjadi, “A multiyear DG-incorporated framework for expansion planning of distribution networks using binary chaotic shark smell optimization algorithm,” *Energy*, vol. 102, pp. 199–215, 2016.
- [36] M. Isiet and M. Gadala, “Self-adapting control parameters in particle swarm optimization,” *Appl. Soft Comput. J.*, vol. 83, p. 105653, 2019.
- [37] N. Singh and S. B. Singh, “A Modified Mean Gray Wolf Optimization Approach for Benchmark and Biomedical Problems,” *Evol. Bioinforma.*, vol. 13, 2017.

and developing Malaysian Standards and IEC standards related to photovoltaic systems. Email: shahril_irwan2004@yahoo.com



Intan Rahayu Ibrahim received Bachelor of Electrical and Electronic Engineering degree (B. Eng (Hons)) in 2000 and Meng. (Electrical) by Research in 2005 from Universiti Tenaga Nasional (UNITEN) Malaysia. She later obtained her PhD in Electrical Engineering from Universiti Teknologi MARA in 2017. She was a lecturer in College of Engineering UNITEN from 2001 to 2007. In 2007, she joined Faculty of Electrical Engineering, Universiti Teknologi MARA Pulau Pinang where she is currently a senior lecturer. Her main research interests are in photovoltaic systems, power electronics and micro coprocessor system. She has broad research experience in design and operation of both grid-connected and off-grid photovoltaic systems and design of power converter system for PV energy system. Apart from that, she is also a graduate engineer of Board of Engineers Malaysia as well as a member of Institute Electrical & Electronics Engineering (IEEE). She has been involved in renewable energy industry in Malaysia as technical expert for Green Technology Financing Scheme (GTFS) and she is certified competent person by Sustainable Energy Development Authority (SEDA) for designing Grid-Connected Photovoltaic (GCPV) system.

AUTHORS PROFILE



Nur Atharah Kamarzaman currently pursuing her PhD in Universiti Teknologi Mara, Malaysia in the area of optimization photovoltaic system. She is a lecturer at Universiti Teknologi Mara, Pulau Pinang, Malaysia. She received her B.Eng (Hons.) from Universiti Teknologi Mara, Malaysia in 2010 and M.Eng (Power) from Universiti Teknologi Malaysia, Malaysia in 2013. Her current interest is in Artificial Intelligence and Renewable Energy.



Shahril Irwan Sulaiman holds a PhD in Electrical Engineering from Universiti Teknologi MARA, Malaysia. He obtained his M.EngSc in Photovoltaic Engineering from University of New South Wales, Australia and B.Eng in Electrical & Electronics from Universiti Tenaga Nasional, Malaysia. He is an Associate Professor in Faculty of Electrical Engineering, Universiti Teknologi MARA, Malaysia and the Head of Green Energy Research Centre at the university. Besides conducting numerous contract-based research, he has been assisting the national photovoltaic industry and the government for more than a decade in areas related to design, installation, testing & commissioning, operation and maintenance of both grid-connected photovoltaic systems and stand-alone photovoltaic systems. He is also a member of working groups in reviewing