Chapter 4
Optimal Allocation of Distributed Energy Resources Using Modern Optimization Techniques

4.1 Metaheuristic Optimization Techniques

4.1.1 Marine Predators Algorithm (MPA)

The marine predators algorithm (MPA) is a modern technique that mimics the marine predators foraging such as tunas, marlines, swordfish, sunfish, and sharks with their motion in oceans [144]. The marine predators foraging phase is based on two motions, comprising the Brownian motion and the Lévy flight walk motion, shown in Fig. 4.1. It is worth mentioning that the Lévy flight walk is an arbitrary process of a transition of an object from one place to another using a probability distribution factor [23].

The predators move in a Brownian movement pattern when these predators exist in a region that has prey abundant. In contrast, the predators move like Lévy flight walk at the food in a prey-sparse environment. Besides that, the predators in sudden vertical jumps are recorded by the Fish Aggregating Device (FAD). The steps procedure and the mathematical representation of the MPA are discussed in the following:

4.1.2 Initialization

The initial populations are generated randomly using (4.1) as follows:

$$X_i = X_i^{\min} + \left(X_i^{\max} - X_i^{\min}\right) \times \text{rand} \tag{4.1}$$

where $X_i^{\max}$ means the maximum limit of variable I, while $X_i^{\min}$ denotes to its minimum limit.

rand$\in$ [0–1]. Then, the fitness function is calculated as follows:

A. M. A. El-Hameid et al., *Enhancement of Grid-Connected Photovoltaic Systems Using Artificial Intelligence*, https://doi.org/10.1007/978-3-031-29692-5_4

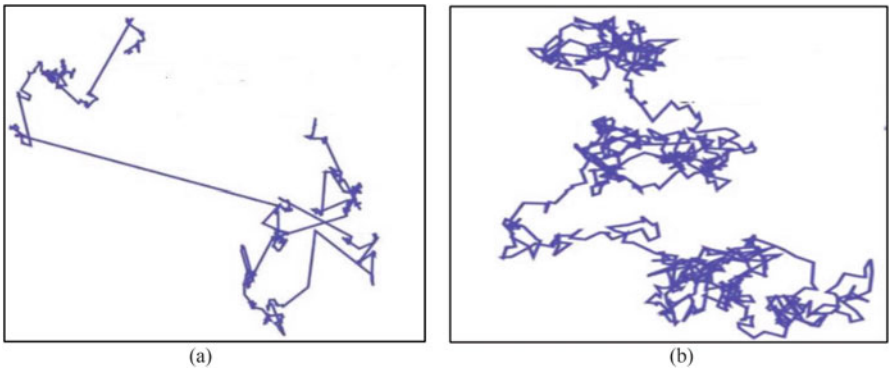

Fig. 4.1 Trajectories (**a**) Lévy flight motion and (**b**) Brownian motion

$$F_i = \text{obj}\ (X_i) \tag{4.2}$$

4.1.3 Assigning the Top Predator

A matrix is constructed to organize the populations which are called the prey matrix:

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1d} \\ X_{2,1} & X_{22} & \cdots & X_{2,1} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \ldots & X_{n,d} \end{bmatrix} \tag{4.3}$$

where n denotes the population number and d is the number of control variables. The top predators are assigned by arrangement the populations based on their fitness functions. The elite matrix is constructed, which represents the top predators:

$$E = \begin{bmatrix} E_{1,1} & E_{1,2} & \cdots & E_{1d} \\ E_{2,1} & E_{22} & \cdots & E_{2,1} \\ \vdots & \vdots & \ddots & \vdots \\ E_{n,1} & E_{n,2} & \ldots & E_{n,d} \end{bmatrix} \tag{4.4}$$

4.1.4 The Brownian and Lévy Flight Orientations

The locations of the predators and the prey are updated using three phases, which are based on the ratio of the velocity between the predator and the prey.

Phase 1: This phase is implemented when the predator's velocity is more than the prey's velocity. The phase mimics the exploration process of the algorithm, and the predator and the prey move like a Brownian movement pattern, which can be formulated as follows:

$$\overline{\mathrm{SEZ}_i} = \overline{R_{\mathrm{Br}}} \bigoplus \left(\overline{E_i} - \overline{R_{\mathrm{Br}}} \bigoplus \overline{X_i}\right) \quad \text{if } T \le T_{\max} \tag{4.5}$$

$$\overline{X_i} = \overline{X_i} + P.\bar{R} \bigoplus \overline{\mathrm{SEZ}_i} \tag{4.6}$$

where T is the current iteration, while $T_{\max}$ represents the maximum number of iterations, respectively; $\bigoplus$ represents the entry wise multiplication. $\overline{R_{\mathrm{Br}}}$ denotes a vector that includes random numbers from the Brownian motion. $\overline{\mathrm{SEZ}_i}$ is a step size vector, while P is a constant number, which is equal to 0.5.

Phase 2: This phase is implemented when the velocities of the predator and the prey are identical. Thus, the book is considered an intermitted phase between the exploitation and the exploration processes. The populations are separated into two sections: the first section is utilized for the exploration, while the second section is applied for the exploitation. Thus, this step can be represented as follows:

$$\text{if } \frac{1}{3}T_{\max} \le T \le \frac{2}{3}T_{\max}$$

$$\overline{\mathrm{SEZ}_i} = \overline{R_{\mathrm{Levy}}} \bigoplus \left(\overline{E_i} - \overline{R_{\mathrm{Levy}}} \bigoplus \overline{X_i}\right) \quad \text{for } i = 1,2,3\ldots, \quad \frac{n}{2} \tag{4.7}$$

$$\overline{X_i} = \overline{X_i} + P.\bar{R} \bigoplus \overline{\mathrm{SEZ}_i} \tag{4.8}$$

The second section is represented as follows:

$$\overline{\mathrm{SEZ}_i} = \overline{R_{\mathrm{Br}}} \bigoplus \left(\overline{E_i} - \overline{R_{\mathrm{Br}}} \bigoplus \overline{X_i}\right) \qquad \text{for } i = \frac{n}{2}, \ldots, n \tag{4.9}$$

$$\overline{X_i} = \overline{X_i} + P.\mathrm{CF} \bigoplus \overline{\mathrm{SZ}_i} \tag{4.10}$$

where R_{Levy} denotes a vector that mimics the motion of prey in Lévy flight motion. CF represents the adaptive operator which is used for controlling the step size of predator motion.

Phase 3: This phase represents the exploitation process used when the predator's velocity is more than the prey's velocity. In this stage, the populations or the predator moves in Lévy flight orientation, which can be represented as follows:

$$\overline{\mathrm{SEZ}_i} = \overline{R_{\mathrm{Levy}}} \bigoplus \left(\overline{R_{\mathrm{Levy}}} \bigoplus \overline{E_i} - \overline{X_i}\right) \quad \text{for } i = 1,2,3\ldots, \quad \frac{n}{2} \tag{4.11}$$

$$\overline{X_i} = \overline{E_i} + P.\bar{R} \bigoplus \overline{\mathrm{SEZ}_i} \tag{4.12}$$

4.1.5 FDAs Effect and Eddy Formation

Most sharks spent about 80% of their time in the closed region, while 20% of their time moves to new areas are known as Fish Aggregating Devices (FADs) or the eddy formation. Representation of this step is formulated as follows:

$$\overline{X_i} = \begin{cases} \overline{X_i} + \mathrm{CF}\left[X_i^{\min} + R\left(X_i^{\max} - X_i^{\min}\right)\right] \bigoplus \bar{U} & \text{If } r \leq \mathrm{FADS} \\ \overline{X_i} + \left[\mathrm{FADS}(1-r) + r\right]\left(\overline{X_{r1}} - \overline{X_{r2}}\right) & \text{If } r > \mathrm{FADS} \end{cases} \tag{4.13}$$

where r denotes a random value $r \in [0–1]$. $r1$ and $r2$ are indices selected from the prey matrix. FADs are the probability of the FADs which is 0.2. $\bar{U}$ represents a binary vector.

4.1.6 Marine Memory

The marine predators remember the optimal location efficiently for foraging. Therefore, in the MPA, the new solution is compared to those in the last iteration to obtain the best solution. Figure 4.2 shows the flowchart of application the MPA for optimal photovoltaic distributed generation (PV-DG) and distributed static compensator (D-STATCOM).

4.2 Equilibrium Optimizer (EO)

EO is a recent physical-based algorithm that mimics the mass balance of a dynamic mixture on a control volume [145]. The mass balance equation consists of input, output, and generated mass in a control volume and can be given as follows:

$$V\frac{\mathrm{d}c}{\mathrm{d}t} = QC_{\mathrm{eq}} - QC + G \tag{4.14}$$

where V represents the control volume, C is the concentration of V, Q is the input and output flow rate, Ceq represents the equilibrium state concentration, and G represents the mass generation rate. By integrating and manipulating Eq. (4.14) the following is obtained:

$$C = C_{\mathrm{eq}} + \left(C_0 - C_{\mathrm{eq}}\right)\exp[-\lambda(t-t_0)] \quad + \frac{G}{\lambda V}\left(1 - \left(\exp[-\lambda(t-t_0)]\right)\right) \tag{4.15}$$

where $\lambda = \left(\frac{Q}{V}\right)$. C_0 represents the initial concentration while t_0 denotes the initial start time. Equation (4.15) is the main updating equation of the EO where the

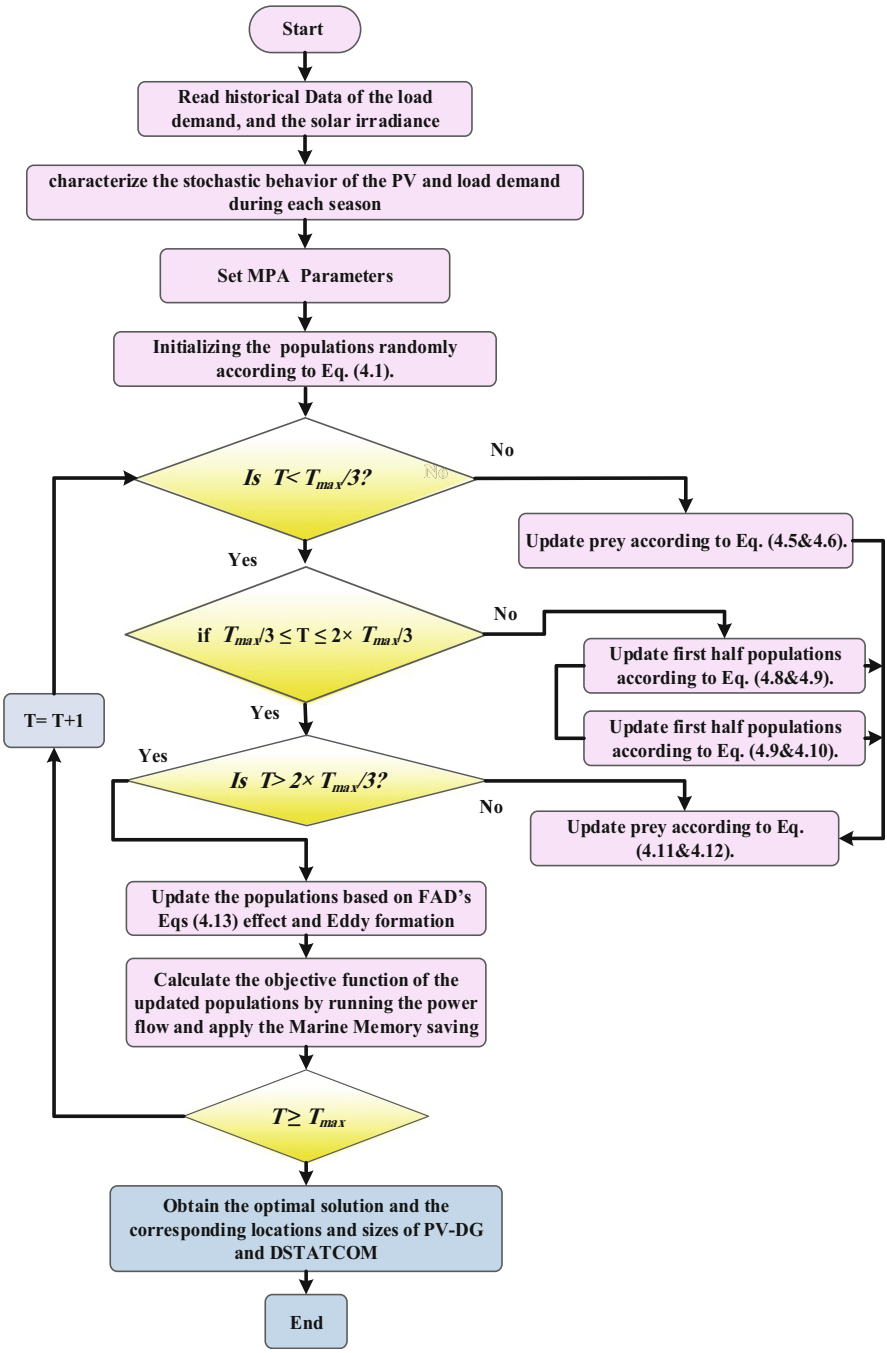

Fig. 4.2 The flowchart of application the MPA for optimal PV-DG and D-STATCOM

concentration represents the particle's position. Equation (4.15) depends on three terms. The first one is related to the equilibrium concentration, the second term denotes the exploration phase of the algorithm, while the third term represents the exploitation phase of the algorithm where it represents the generation rate, which enables the solution to move in small steps. The steps of the EO can be summarized as follows:

Step 1: Initialization
Obtain the initial particles (concentrations) randomly according to (4.16)

$$C_i^{\text{initial}} = C_{\min} + \text{rand}_i\,(C_{\max} - C_{\min}) \qquad i = 1, 2, \ldots\ldots n \tag{4.16}$$

where $C_{\min}$ and $C_{\max}$ represent the minimum and maximum values of the control variables. rand_i is the random variable within [0,1]. Then, calculate the objective function for the initial points.

Step 2: Assigning the Equilibrium Candidates
The particles are sorted based on their corresponding values, and the best four particles are averaged as illustrated in (4.17).

$$\vec{C}_{\text{eq(avg)}} = \frac{\vec{C}_{\text{eq1}} + \vec{C}_{\text{eq2}} + \vec{C}_{\text{eq3}} + \vec{C}_{\text{eq4}}}{4} \tag{4.17}$$

Then, equilibrium pool vector is constructed which includes the five particles which are the four best particles, and the averaged particles are as follows:

$$\vec{C}_{\text{eq,pool}} = \left\{ \vec{C}_{\text{eq1}}, \vec{C}_{\text{eq2}}, \vec{C}_{\text{eq3}}, \vec{C}_{\text{eq4}}, \vec{C}_{\text{eq(avg)}} \right\} \tag{4.18}$$

Step 3: Updating the concentration
The concentrations are updated exponentially with iteration based on exponential term (F) which is based on the control volume λ, where it represents a random vector within the range [0, 1]:

$$\vec{F} = a_1 \text{sign}\left(\vec{r} - 0.5\right)\left[e^{-\vec{\lambda} t} - 1\right] \tag{4.19}$$

$$t = \left(1 - \frac{T}{T_{\text{Max}}}\right)^{\left(a_2 \frac{T}{T_{\text{Max}}}\right)} \tag{4.20}$$

where a_1 and a_2 are constant values utilized to adjust the exponential value and they are selected to be 2 and 1, respectively, and T and T_{Max} present the current and the maximum number of iterations, respectively. It should be highlighted here that a_1 controls the exploration quantity of the algorithm while $a2$ controls the exploitation feature of the algorithm. Sign $(r - 0.5)$ controls the exploration direction.

Step 4: Update of the concentration depends upon the generation rate.
The populations are updated in this step by applying the generation rate method for enhancing the exploitation process of this algorithm which can be described as follows:

$$\vec{G} = \overrightarrow{G_0}\, e^{-\vec{k}\,(t - t_0)} \tag{4.21}$$

$$\overrightarrow{G_0} = \overrightarrow{\mathrm{GCP}}\, \left(\overrightarrow{C_{\mathrm{eq}}} - \vec{\lambda}\, \vec{C}\right) \tag{4.22}$$

$$\vec{\mathrm{GCP}} = \begin{cases} 0.5\, r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \tag{4.23}$$

where r_1 and r_2 are random numbers in [0, 1] and GCP vector is constructed by the repetition of the same value resulted from. GP is the generation probability that controls the participation probability of concentration updating by the generation rate. GP = 1 means that there will be no generation rate term participating in the optimization process. GP = 0 means that the generation rate term will be participating in the process. GP = 0.5 provides a good balance between exploration and exploitation phases.

The updated equation of EO based on the previous steps can be formulated as follows:

$$\vec{C} = \overrightarrow{C_{\mathrm{eq}}} + \left(\vec{C} - \overrightarrow{C_{\mathrm{eq}}}\right).\vec{F} + \frac{\vec{G}}{\vec{\lambda}\, V}\left(1 - \vec{F}\right) \tag{4.24}$$

Step 5: Adding memory saving.
In this step, the concentration will track the obtained best coordination where the objective function of each particle is compared with its value in the previous iteration. A flowchart of the EO with sensitivity analysis for allocation of PV and D-STATCOM is illustrated in Fig. 4.3.

The procedure of application the EO for optimal allocation the PV unit and D-STATCOM can summarized as follows.

Step 1: Set the parameters of the EO and the system data including the line data and the bus data.
Step 2: Run the load flow and carry out the sensitivity analysis of the system and assign the high potential buses for allocation the PV unit and D-STATCOM.
Step 3: Initialize the populations randomly.
Step 4: Calculate the objective function of the obtained populations by running the load flow.
Step 5: Determine the equilibrium candidates (the best four solution and their average values).
Step 6: Construct F, G0, and GCP according to (4.19), (4.22), and (4.23), respectively.

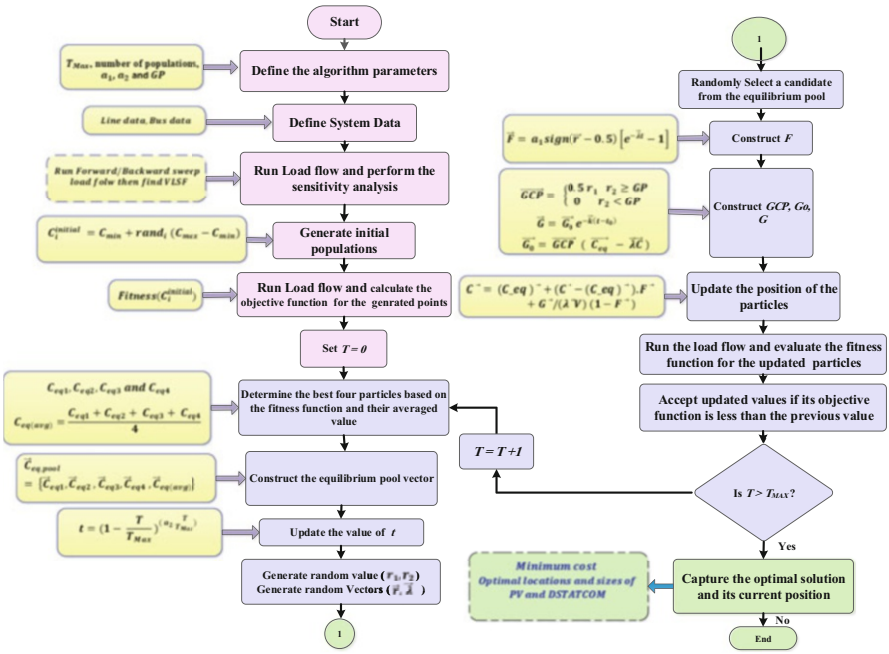

Fig. 4.3 Solution process of EO and sensitivity analysis for allocation of PV and D-STATCOM

Step 7: Update the locations of the population according to (4.24).
Step 8: Accept the new updated population its fitness function is better that the previous solution.
Step 9: Go to step (4) if the stopping criteria is not stratified. Otherwise, end the program and obtain the best solution (the optimal locations and ratings of the PV and D-STATCOM and the corresponding fitness function).

4.3 Lightning Attachment Procedure Optimization (LAPO)

LAPO is a novel physical-based algorithm that mimics the formation of lightning in nature. Lightning happens when a large amount of the positive and negative charges accumulate in the cloud. When the amount of these charges increases, the electric voltage will increase leading to air breakdown. Consequently, the lightning occurs in several paths including the downward and upward leaders. Figure 4.4 depicts the formation of the positive and negative charges inside the cloud as well as the formation of the downward leader and the upward leaders. Referring to this figure, the lightning starts from several points which are known as the initial spots of the leaders. Also, it can be obvious that the direction of the upper ward leaders and the downward leaders are in opposite direction, this sequence leads to formation of

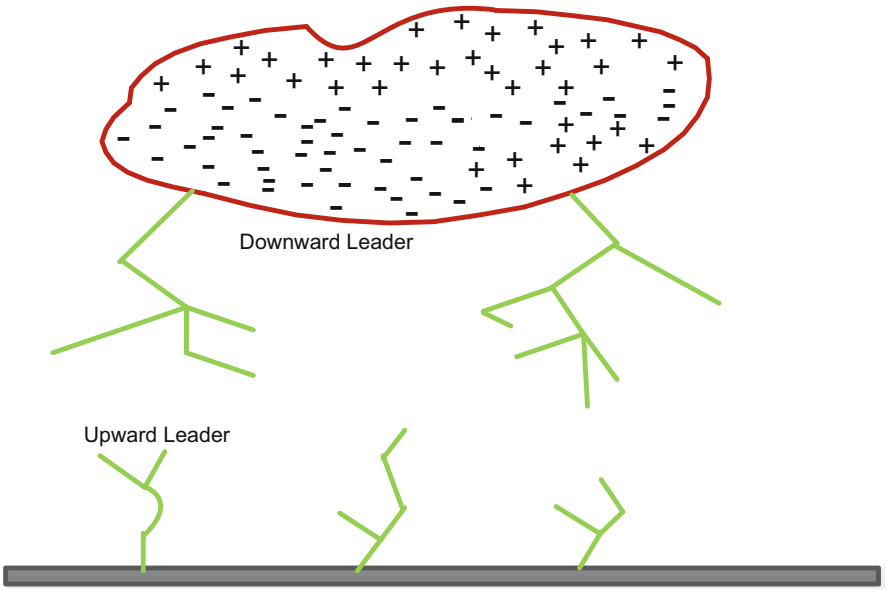

Fig. 4.4 Formation of the charges and leaders in the cloud

lightning in nature. The lightning is based on four steps which are (1) breakdown of air at the surface of the cloud, (2) motion of the downward leader, (3) the motion of the upward leader, and (4) strike location. The mathematical representation of the LAPO algorithm can be depicted as follows:

Step 1: Initialization
The initial points or the trial spots of the downward leaders according to Eq. (4.25) are as follows:

$$X_{\text{ts}}^{i} = X_{\text{min}}^{i} + \left(X_{\text{max}}^{i} - X_{\text{min}}^{i}\right) \times \text{rand} \tag{4.25}$$

where X_{ts}^{i} is the trial spots. X_{min} denotes the minimum boundary of the control variables, while X_{max} is the maximum value. Rand represents a random value within range [0,1]. Then, find the objective function of the initial points:

$$F_{\text{ts}}^{i} = \text{obj}\left(X_{\text{ts}}^{i}\right) \tag{4.26}$$

Step 2: The next jump determination
The downward leader points are averaged and the objective function of this value is evaluated as follows:

$$X_{\text{avr}} = \text{mean}\ (X_{\text{ts}}) \tag{4.27}$$

$$F_{\text{avr}} = \text{obj}\ (X_{\text{avr}}) \tag{4.28}$$

X_{avr} denotes the averaged point. F_{avr} represents the averaged objective function point. For updating point i, a random solution j is selected where $i \neq j$. Then, the updated solution will be compared with the selected solution. Then, the next jump is evaluated as follows.

$$X^i_{\text{new}} = X^i_{\text{ts}} + \text{rand} \times \left(X_{\text{avr}} + X^j_{\text{PS}}\right) \qquad \text{IF}\ \ F_j < F_{\text{avr}} \tag{4.29}$$

$$X^i_{\text{new}} = X^i_{\text{ts}} - \text{rand} \times \left(X_{\text{avr}} + X^j_{\text{PS}}\right) \qquad \text{IF}\ \ F_j > F_{\text{avr}} \tag{4.30}$$

Step 3: Section Diminishing
Section Diminishing mean pick the new solution if it better than the original solutionwhich can be considered as follows.

$$X^i_{\text{ts}} = X^i_{\text{new}} \qquad \text{IF}\ \ F^i_{\text{new}} < F^i_{\text{ts}} \tag{4.31}$$

$$X^i_{\text{new}} = X^i_{\text{ts}}\ \text{otherwise} \tag{4.32}$$

Step 4: Motion of the upward leader
In this step, the obtained Value from the previous solution is updated based on exponent operator, the best solution, and the worst solution as follows:

$$X^i_{\text{new}} = \ X^i_{\text{new}} + \text{rand} \times S \times \left(A^i_{\text{best}} - A^i_{\text{worst}}\right) \tag{4.33}$$

where

$$S = 1 - \left(\frac{t}{t_{\max}}\right) \times e^{\left(-\frac{t}{t_{\max}}\right)} \tag{4.34}$$

where t represents the current iteration, $t_{\max}$ represents the maximum number of iterations, and X^i_{best} and X^i_{worst} denote the best and the worst solution, respectively.

Step 5: Strike point
Strike point is the point of combination of the downward and upward leaders which represent the optimal solution. Flowchart of the LAPO with sensitivity analysis for allocation of PV and D-STATCOM is illustrated in Fig. 4.5.

The procedure of application the LAPO for optimal allocation the PV unit and D-STATCOM can be summarized as follows:

Step 1: Set the parameters of the LAPO as well as the system data.
Step 2: Run the load flow and carry out the sensitivity analysis of the system and assign the high potential buses for allocation the PV unit and D-STATCOM.

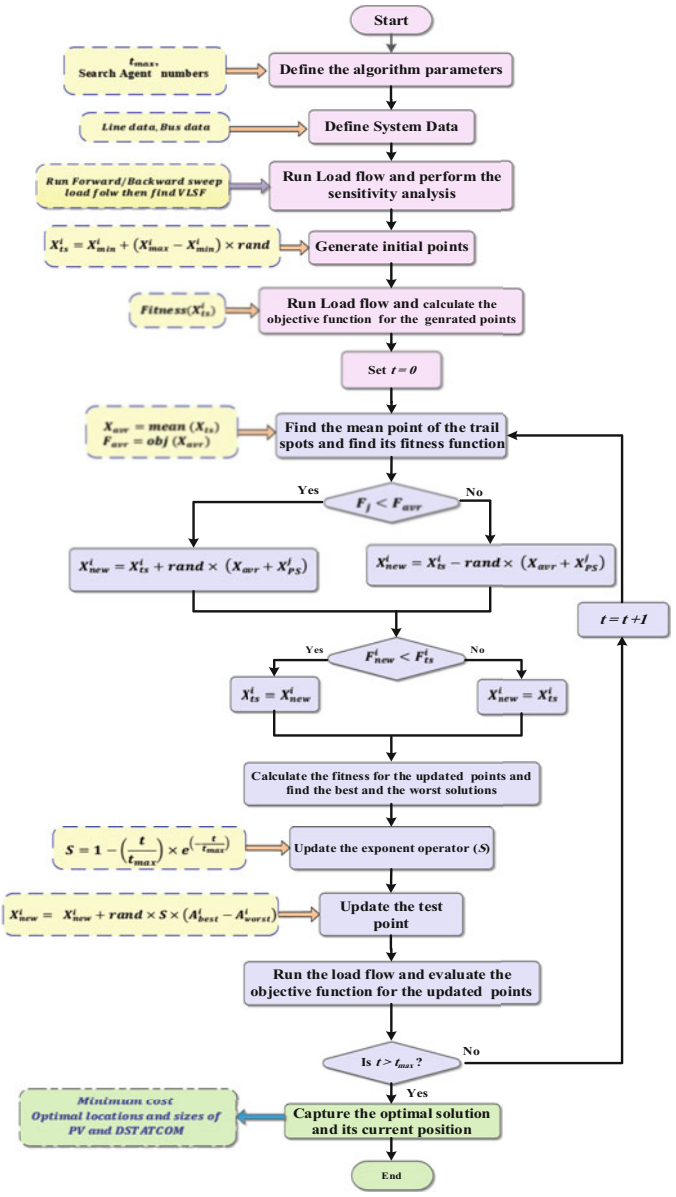

Fig. 4.5 Solution process of LAPO and sensitivity analysis for allocation of PV and D-STATCOM

Step 3: Initialize the spot points randomly.
Step 4: Run the load flow and evaluate the objective function of the obtained points.
Step 5: Find the average vector point of the generated points and its corresponding objective function.

Step 6: Update the location of the points using (4.29) and (4.30), respectively.
Step 7: Run the load flow and calculate the objective function of the updated points and accept the new updated population its fitness function is better that the previous solution.
Step 8: Update the locations of the points using (4.33). Then, accept or reject the new updated point based on their objective functions.
Step 9: Go to step (4) if the stopping criteria is not stratified. Otherwise, end the program and obtain the best solution (the optimal locations and ratings of the PV and D-STATCOM and the corresponding fitness function).

4.4 Sine Cosine Algorithm (SCA)

SCA is an efficient algorithm which is conceptualized from the sine and cosine function trends [27]. The orientation of the search agents around the best solution with iterative process based on sine cosine trends is depicted in Eq. (4.35) as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + C_1 \times \sin(C_2) \times \left| C_3 X_{\text{best}}^t - X_i^t \right| & C_4 < 0.5 \\ X_i^t + C_1 \times \cos(C_2) \times \left| C_3 X_{\text{best}}^t - X_i^t \right| & C_4 > 0.5 \end{cases} \tag{4.35}$$

where t denotes the iteration number. X_{best}^t denotes the best location from the search agents. C_2, C_3, and C_4 denotes to random variables within [0, 1]. C_1 represents an adaptive factor that can be given as follows:

$$C_1 = k - t\frac{K}{T_{\max}} \tag{4.36}$$

where k is a fixed value. $T_{\max}$ is the maximum number of iterations. Equation (4.36) characterize the main countenance of the SCA algorithm as shown in Fig. 4.6 which shows the variability of the position of the groups around the best situation based on the size and phase angle difference of the sine and cosine functions. y_1 adjusts the new criterion placement to move outwardly or inwardly in the best placement as shown in Fig. 4.7.

4.4.1 Enhanced Sine Cosine Algorithm (ESCA)

ESCA is based on Lévy flight distribution (LFD); LFD is integrated into SCA technology to foster the search ability and reconnaissance capacity of this optimization algorithm by rising the likelihood of manufacture new solutions to avoid algorithm stagnation and to avert trapping at local minimums. Lévy's trip is a random operation for generating a new solution based on a random walk whose

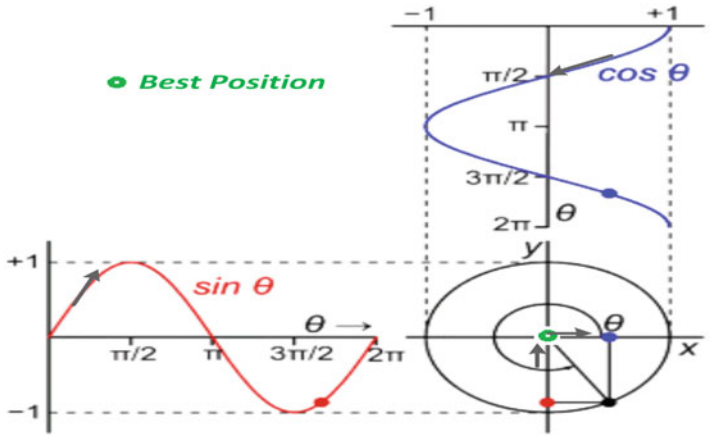

Fig. 4.6 Inhabitance movement about the best solution based on cosine

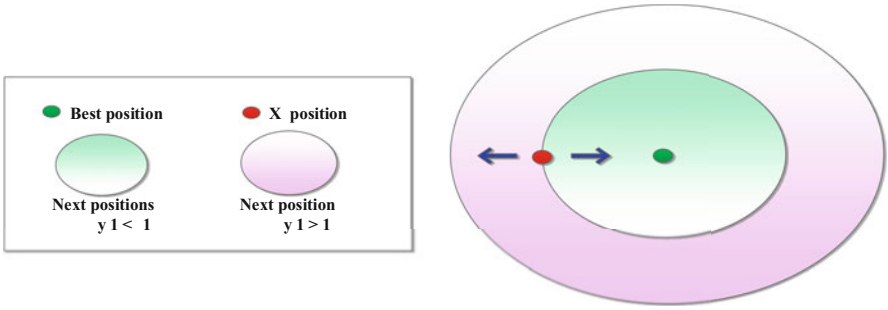

Fig. 4.7 The trend of the following status on the best placement counts on y_1

steps are captured from the Lévy distribution. The new inhabitance situations which count on a fibrous distribution can be found as follows. ESCA is a new version of the SCA to overcome its stagnation. In the ESCA, the Lévy flight distribution (LDF) is utilized to improve the algorithm exploration ability where it enables the populations to jump to new areas. The new population position that is based on Lévy distribution can be found as follows [28]:

$$X_i^{\text{new}} = X_i + \propto \bigoplus \text{Levy}\ (\beta) \tag{4.37}$$

where $\propto$ denotes a random step size factor. $\bigoplus$ represents the multiplication of entry wise, while β denotes an LDF. The step size is calculated given as:

$$\propto \bigoplus \text{Levy}(\beta) \sim 0.01 \frac{w}{|v|^{1/\beta}} \left(X_i^t - X_{\text{best}}^t\right) \tag{4.38}$$

where u and v are usually variables produced by a normal distribution where

$$w \sim n\left(0, \phi_u^2\right), v \sim n\left(0, \phi_v^2\right) \tag{4.39}$$

$$\phi_{\mathrm{w}} = \left[\frac{\Gamma(1+\beta) \times \sin(\pi \times \beta/2)}{\Gamma[(1+\beta)/2] \times \beta}\right]^{1/\beta}, \phi_v = 1 \tag{4.40}$$

where Γ is the criterion gamma function and $0 \leq \beta \leq 2$. To foster the utilization of SCA, the best search agent is updated by using changing band width as follows:

$$X_{\mathrm{best}}^{\mathrm{new}} = X_{\mathrm{best}}^{t} \pm y_5 \times K_{\mathrm{w}} \tag{4.41}$$

where y_5 is a random number in [0, 1]. K_{w} is a variable bandwidth that decreases dynamically as:

$$K_{\mathrm{w}} = K_{\max} e^{(E \times t)} \tag{4.42}$$

$$E = \left(\frac{\ln\left(\frac{K_{\min}}{K_{\max}}\right)}{T_{\max}}\right) \tag{4.43}$$

where $K_{\max}$ and $K_{\min}$ are the upper and lower sect width limits. t is the current iteration and $T_{\max}$ is the maximum number of iterations. The flowchart of ESCA is shown in Fig. 4.8.

4.5 Ant Lion Optimizer (ALO)

Ant lion optimizer (ALO) is a population-based optimization technique presented by Seydali Mirjalili in 2015 [146]. ALO simulates the hunting behavior of antlion and the interaction between the prey or the ants and the predator antlions where the antlions build circular traps to hunt the ants. The ants move in a stochastic pattern to search for their foods. The mathematic model of the stochastic movement of the ants is formulated as follows:

4.5.1 Random Movement of an Ant

The random movement of the ant is described using the following equation:

$$X(t) = [0, \mathrm{cumsum}(2r(t_1) - 1), \mathrm{cumsum}(2r(t_2) - 1), \mathrm{cumsum}(2r(t_n) - 1)] \tag{4.44}$$

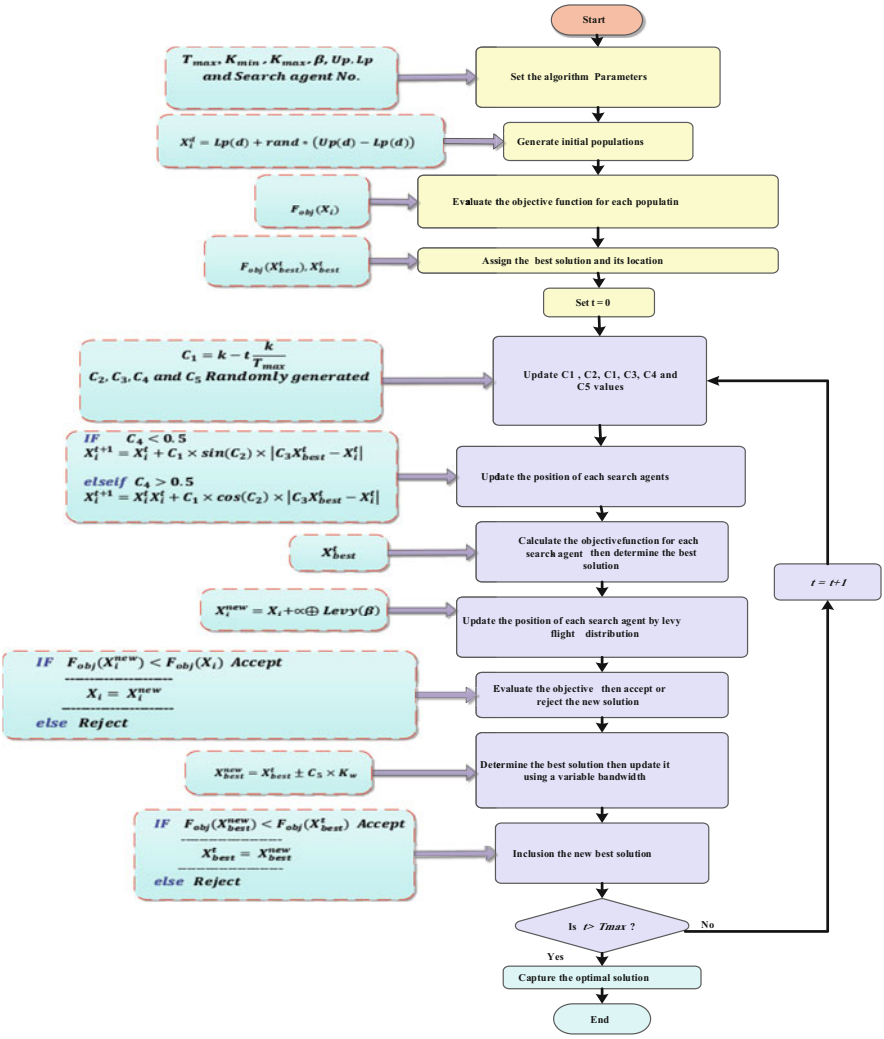

Fig. 4.8 Flowchart of ESCA

where $x(t)$ represents the location of the ant at the t-th iteration. cumsum denotes the cumulative sum. N is the maximum number of iterations. $r(t)$ denotes a random which can be given as follows:

$$r(t) = \begin{cases} 1 & \text{if rand} > 0.5 \\ 0 & \text{if rand} \leq 0.5 \end{cases} \tag{4.45}$$

The min–max normalization equation is utilized for keeping the ants move in random walks inside the search spaces, which can be described as follows:

$$R_i^t = \frac{\left(X_i^t - a_i\right) \times \left(U_i^t - L_i^t\right)}{d_i - a_i} + L_i \tag{4.46}$$

where R_i^t denotes the location of the i-th ant after random walk closed to j-th antlion. U_i^t and L_i^t are the upper and the lower boundaries of i-th variable at t-th iteration, respectively. a_i and d_i represent the minimum and the maximum steps of random walk, respectively. The ants will update their positions based on a random walk, and they will be trapped in the antlion pit. The positions of ants are listed in a matrix as follows:

$$M_{\text{Ant}} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1d} \\ X_{2,1} & X_{22} & \cdots & X_{2,1} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \ldots & X_{n,d} \end{bmatrix} \tag{4.47}$$

The corresponding objective functions for each vector of the ant positions are listed as follows:

$$\text{Obj}_{\text{Ant}} = \begin{bmatrix} f_1(X_{1,1}, X_{1,2}, \ldots, X_{1d}) \\ f_2(X_{1,1}, X_{1,2}, \ldots, X_{1d}) \\ \vdots \\ f_n(X_{1,1}, X_{1,2}, \ldots, X_{1d}) \end{bmatrix} \tag{4.48}$$

The search agents (ant positions) are sorted, and the best agents are selected as antlions, which are listed as follows:

$$M_{\text{Antlion}} = \begin{bmatrix} AL_{1,1} & AL_{1,2} & \cdots & AL_{1d} \\ AL_{2,1} & AL_{22} & \cdots & AL_{2,1} \\ \vdots & \vdots & \ddots & \vdots \\ AL_{n,1} & AL_{n,2} & \ldots & AL_{n,d} \end{bmatrix} \tag{4.49}$$

4.5.2 Trapping in Antlion Pits

The effects antlions' traps on the ant movement can be mathematically represented as follows:

$$L_i^t = \text{Antlion}_j^t + L_i^t \tag{4.50}$$

$$U_i^t = \text{Antlion}_j^t + U_i^t \tag{4.51}$$

4.5.3 Sliding Ants Toward Antlions

When the ants are trapped in the pit of the antlion, the upper and the lower bound should reduce with the increase of iteration as follows:

$$U_i^t = \frac{U_i^t}{I} \tag{4.52}$$

$$L_i^t = \frac{L_i^t}{I} \tag{4.53}$$

where I represents a ratio that can be described as follows:

$$I = 10^{\omega} \frac{t}{T} \tag{4.54}$$

where T and t denote the maximum number of iterations and the current iteration. ω represents a constant which can be described as follows:

$$\omega = \begin{cases} 2 & t > 0.1T \\ 3 & t > 0.5T \\ 4 & t > 0.75T \\ 5 & t > 0.9T \\ 6 & t > 0.95T \end{cases} \tag{4.55}$$

4.5.4 Elitism

The elitism in optimization algorithm means the best solution (best antlion) is saved as an elite, which guide the motion of the populations in the iteration process and can be formulated as follows:

$$X_j^t = \frac{R_{\mathrm{A}}^t + R_{\mathrm{E}}^t}{2} \tag{4.56}$$

where R_{A}^t denotes the random walk closed to the best antlion using the roulette wheel for t-th iteration. R_{E}^tdenotes the position of randomly walking of the j-th ant, nearby the best or the elite antlion (E) in the swarm of ants.

4.5.5 Catching Prey and Rebuilding the Pit

The final stage of hunting behavior of antlions is catching an ant that reaches the pit's bottom. It must update its position to the latest position by the following equation:

$$\text{Antlion}_j^t = \text{Ant}_j^t \text{ if } f\left(\text{Ant}_j^t\right) > f\left(\text{Antlion}_j^t\right) \tag{4.57}$$

4.6 Modified Ant Lion Optimizer (MALO)

MALO is based on enhancing the basic ALO's searching capability by improving the exploration and the exploitation process. The exploration phase is enhanced by applying Lévy flight distribution (LFD), enabling the algorithm to jump to new areas to avoid the basic ALO's stagnation.

$$X_i^{\text{new}} = X_i + \propto \bigoplus \text{Levy}(\beta) \tag{4.58}$$

where $\propto$ denotes a random step parameter. $\bigoplus$ denotes the entry-wise multiplication. β is a parameter related to the LFD? The step size is given as:

$$\propto \bigoplus \text{Levy}(\beta) \sim 0.01 \frac{u}{|v|^{1/\beta}} \left(X_i^t - \text{Antlion}_j^t\right) \tag{4.59}$$

where u and v denote variables obtained by normal distribution as.

$$u \sim N\left(0, \phi_u^2\right), \text{v} \sim N\left(0, \phi_v^2\right) \tag{4.60}$$

$$\phi_u = \left[\frac{\Gamma(1+\beta) \times \sin(\pi \times \beta/2)}{\Gamma[(1+\beta)/2] \times \beta}\right]^{1/\beta}, \phi_v = 1 \tag{4.61}$$

where Γ represents the standard gamma function. $0 \leq \beta \leq 2$. The exploitation process of the algorithm is enhanced by updating the location of ants around the elite (best) solution in a spiral path as follows:

$$X_i^{\text{new}} = \left|\text{Antlion}_j^t - X_i\right| e^{bt} \cos(2\pi t) + \text{Antlion}_j^t \tag{4.62}$$

b is a constant used to define the logarithmic spiral shape. To balance between the exploitation and the exploration, an adaptive operator is used for this sake as depicted in the following.

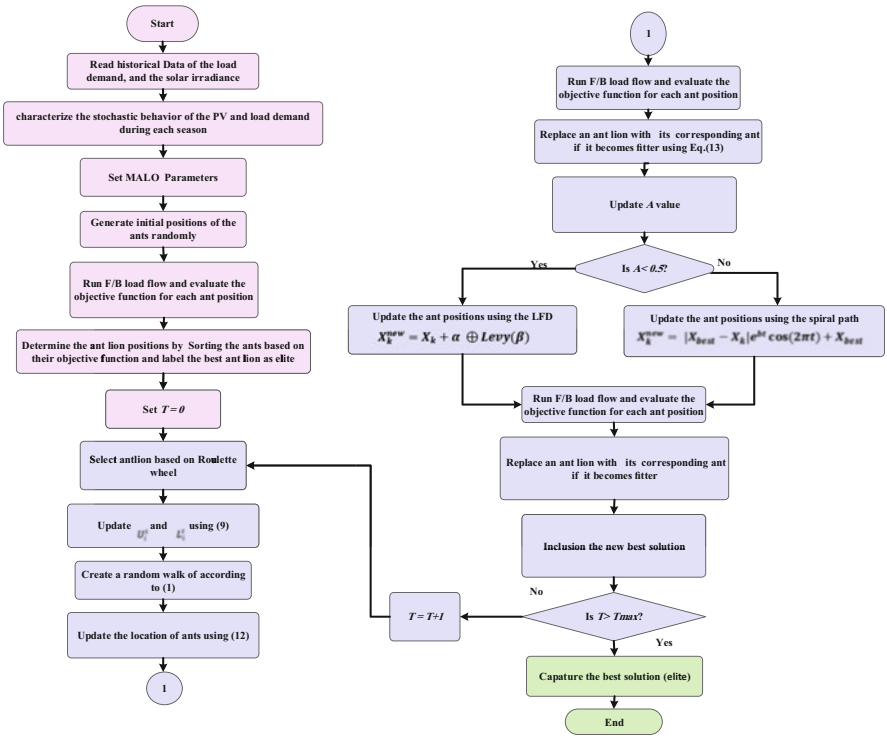

Fig. 4.9 Flowchart of application of the MALO for optimal planning

$$A(t) = A_{\min} + \left(\frac{A_{\max} - A_{\min}}{T}\right) \times t \tag{4.63}$$

where $A_{\max}$ and $A_{\min}$ are the maximum and the minimum A limits. This value is changed dramatically from $A_{\max}$ to $A_{\min}$. When the value of A is closed to$A_{\min}$, the position of the populations will be updated using (4.57) it while enhancing the exploration of this technique while when the value of A is closed to $A_{\max}$, the position of the populations will be updated using (4.61), which enhance the exploitation of this technique. The application of the proposed algorithm for solving the planning problem is depicted in Fig. 4.9.

4.7 Whale Optimization Algorithm (WOA)

4.7.1 Inspiration

The whale is considered the biggest animal on earth, even by its height or weight. Its length can exceed 30 meters and its weight can exceed 180 tons. Whales are

categorized into seven species: killer, minke, blue, sei, humpback, right, and finback. It is worth mentioning that whales have high emotions and feelings above many other animals [147]. Wales has a special brain formation, where there are spindle cells similar to human brain cells with a double number compared to a human. Spindle cells enable whales to think, judge, and create social relationships in a smart way but at a low level compared to human [147].

Whales may live individually or in groups. Killer spends all their lifetime in groups. Humpback whales have huge baleen and their size is nearly the same as a school bus for the adult one. Humpback whales have a special method for hunting their food which includes krill and small fish herds; this hunting method is called bubble-net feeding [148]. Hunting is achieved nearby to the water surface by forming a very big quantity of bubbles in the form of nine-shaped path [149]. The hunting method is accomplished in two maneuvers "upward spirals" and "double-loops." Humpback whales go downward for nearly 12 meters then initiate producing bubbles in the form of spirals surrounding the prey then they start to go upward to the water surface. WOA is represented mathematically as explained in the following paragraph: upward leader and (4) strike location. The mathematical representation of the WOA algorithm can be depicted as follows:

4.7.2 Mathematical Model and Optimization Algorithm

Mathematical representation for hunting process can be explained as follows.

4.7.2.1 Circling Prey

The first step of the hunting process is to assign the position and surround the prey. Position of the prey is considered as the best solution which is nearby to the global solution within the search space. The mathematical equation representing this behavior can be formulated in the following equations.

$$\vec{D} = \left(\left| \vec{C} . \overrightarrow{X^*}(t) - \overrightarrow{X^*}(t) \right| \right) \tag{4.64}$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A} . \vec{D} \tag{4.65}$$

t denotes the current iteration

A and C represent vectors coefficient

X^* represents the position of best solution

X represents the position vector

The resulted of the best solution X^* is characterized by updating its value for each iteration until reaching to global optima or the best solution.

$\vec{A}$ and $\vec{C}$ can be calculated from the following equations.

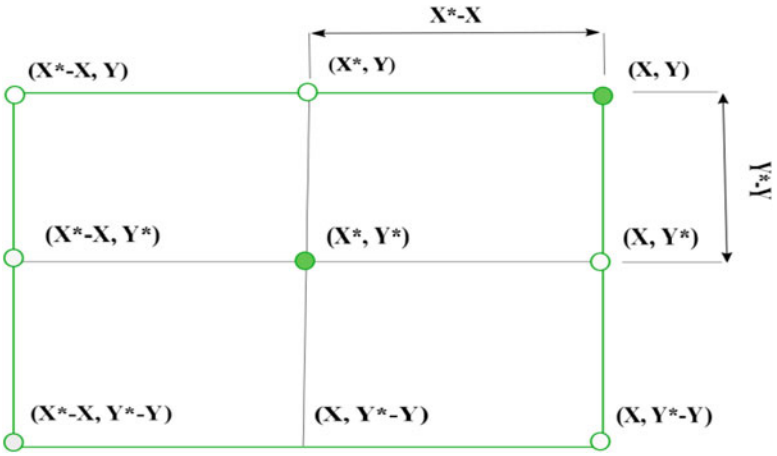

Fig. 4.10 Position vectors in 2D along with the expected positions

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{4.66}$$

$$\vec{C} = 2 \cdot \vec{r} - \vec{a} \tag{4.67}$$

where $\vec{a}$ is a coefficient that decreases its value within the search space from 2 to 0 through proceeding of the process, and $\vec{r}$ is a vector that varies randomly within the period [0,1]. Fig. 4.10 shows the 2D relationship between the current positions of candidate (X, Y) and the best record (X^*, Y^*) according to vectors $\vec{A}$ and $\vec{C}$ values.

Equation (4.65) represents the mathematical formulation of encircling the prey by the search agents inside the search space. For n dimensions search space, the same idea can be applied. The candidates are moving in hypercube around the best solution. The second stage of the hunting process is explained as follows:

4.7.2.2 Bubble-Net Attacking Method

This method of humpback whales for hunting process represents the exploitation phase can be applied according to two mechanisms; this foraging is done by creating distinctive bubbles along a circle or "nine"-shaped path as shown in Fig. 4.11, which forms a curtain and frights the prey for getting them together to speed up the efficiency of hunting [148, 150].

4.7.2.3 Shrinking Encircling Mechanism

In this mechanism, the vector $\vec{a}$ decreases according to (4.66). The vector $\vec{A}$ changes its value within the interval $[-a, a]$ which descending from 2 to 0. If the

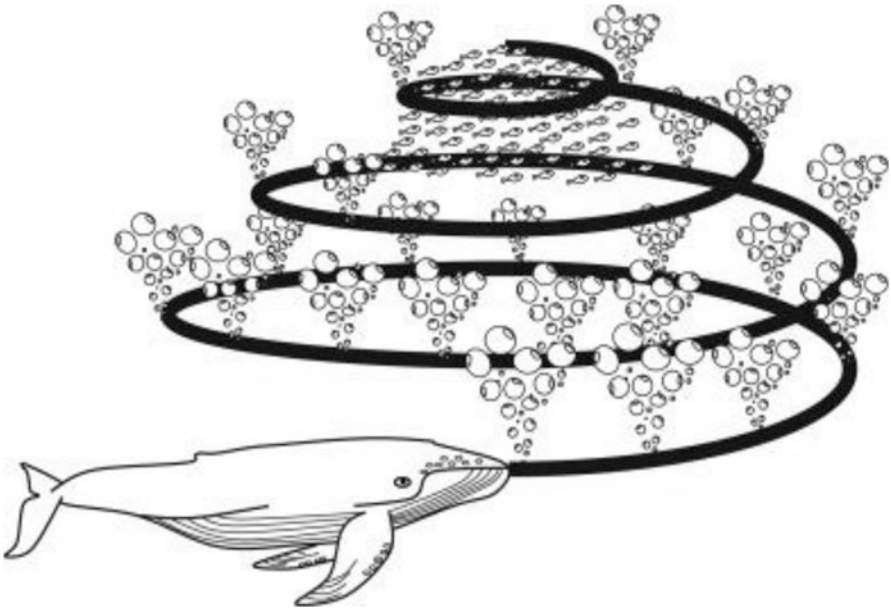

Fig. 4.11 Spiral bubble-net behavior of humpback whales

interval of $\vec{A}$ is set to $[-1, 1]$, the search agents will update their locations between the original position and the new location of the best agent, the relative locations of the (X, Y) with respect to $(X*, Y*)$ lies within $0 \leq A \leq 1$ when the search space is 2D.

4.7.2.4 Spiral Updating Position

The spiral path of humpback whales is depicted in the mathematical equations of this mechanism and can be formulated as follows.

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) \tag{4.68}$$

$$\overrightarrow{D'} = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{4.69}$$

where D' is the distance from the prey to the humpback whale (best solution), b is the shape coefficient for the spiral movement, l is a number vary randomly within the interval $[-1, 1]$, and "." represents that to multiply element by element. The spiral movement of hunting process is reduced in the form of two shapes in the same time. It can be considered as a probability of 50% for choice between the spiral shape and the encircling mechanism as follows.

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A}.\vec{D} & \text{if } p < 0.5 \\ \vec{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) & \text{if } p \geq 0.5 \end{cases} \tag{4.70}$$

The subscript p is a number vary randomly within the interval $[-1, 1]$. The humpback whales' hunting process can be achieved randomly beside the bubble net method. Random search is explained in the following.

4.7.2.5 Search for Prey

Search for the prey represents the exploration phase based on changing the value of vector $\vec{A}$. Random search of the humpback whales depends upon the position of the whales with respect to each other. The value of vector $\vec{A}$ can be changed around 1.0 to oblige the prey to start moving far from the reference whale. The exploitation phase depends upon updating the position of search agents according to the best agent obtained so far. However, the exploration phase depends upon updating the position of search agents randomly. According to this process and $\vec{A} > 1$, exploration phase is ensured and the WOA algorithm can reach to global optima. Representation of this model is formulated in the following eqs. [150].

$$\vec{X}(t+1) = \left| \vec{C}.\overrightarrow{X_{\text{rand}}}(t) - \vec{X} \right| \tag{4.71}$$

$$\vec{X}(t+1) = \overrightarrow{X_{\text{rand}}} - \vec{A}.\vec{D} \tag{4.72}$$

The subscript $\overrightarrow{X_{\text{rand}}}$ denotes a random position vector that refers to randomly chosen whale from the search agents (Fig. 4.12).

4.8 Slime Mold Algorithm (SMA)

This algorithm simulates a method for finding multiple heads in a velarium that inhabits cold and wet places. In this technique, weights approach to negative and positive feedback generated by sticky mold during the foraging process. In this stage, the slurry can determine the best food-gathering path in a superior way. The organic matter in the slime mold searches for food. Then, it encircles it and releases enzymes for its consumption. In the migration stage, the anterior end expands into a fan shape along with a venous network that allows the cytoplasm to slide inward. An intravenous network consists of using multiple food sources simultaneously to form a connection between them. In this mechanism, a reproductive wave is formed when a vein approaches a food source. The mathematical representation of MSA is formulated as follows:

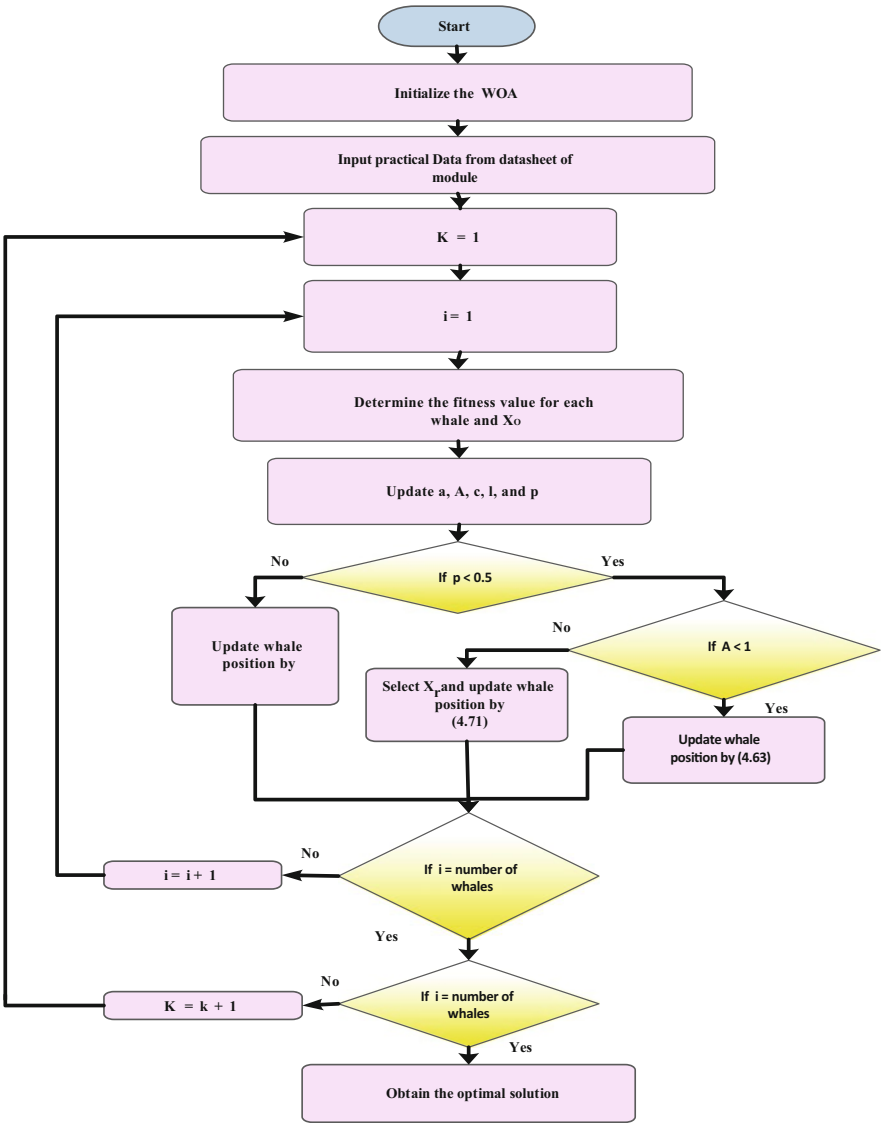

Fig. 4.12 Flowchart of WOA optimization technique

4.8.1 Approach Food

The shrinkage mode of a slime mold can be represented as follows [151]:

$$X(t+1) = \begin{cases} X_b(t) + vb \cdot (W \cdot X_{\mathrm{A}}(t) - X_{\mathrm{B}}(t)), r < p \\ vc \cdot X(t), r \geq p \end{cases} \tag{4.73}$$

where vb denotes a parameter with a range of $[-a, a]$, vc is reduced from one to zero, t denotes the current iteration, X_b is the location of slime mold with the highest odor concentration currently assigned, X denotes the location of slime mold, X_A and X_B are two individuals randomly selected from the swarm, and W represents the weight of slime mold. The p is a parameter which can be obtained as follows:

$$p = \tan h|S(i) - \mathrm{DF}| \tag{4.74}$$

where $i \in 1, 2, \ldots, n$, $S(i)$ represents the fit of $\vec{X}$ and DF represents the best fit obtained in all iterations. The formula of $\overrightarrow{vb}$ is as follows:

$$vb = [-a, a] \tag{4.75}$$

$$a = \arctan h\left(-\left(\frac{t}{\max\_t}\right) + 1\right) \tag{4.76}$$

The W is depicted using (4.77) as follows:

$$W = \begin{cases} 1 + r \cdot \log\left(\dfrac{bF - S(i)}{bF - wF} + 1\right), \text{condition} \\ 1 - r \cdot \log\left(\dfrac{bF - S(i)}{bF - wF} + 1\right), \quad \text{others} \end{cases} \tag{4.77}$$

$$\mathrm{SmellIndex} = \mathrm{sort}(S) \tag{4.78}$$

where condition indicates that $S(i)$ classifies the first half of the population, r denotes the random value in the interval of [0, 1], bF denotes the optimal fitness obtained in the current iterative process, and wF denotes the worst fit it value that has been getting it in the iterative process. Currently, smell index refers to a sequence of ranked fitness values (escalating into the minimum value problem).

4.8.2 Wrap Food

The case simulates a slime mold to control research patterns related to food quality. If the food concentration is contained, the weight near the area is greater; when the concentration of food is low, the weight of the area will decrease, and thus, it will turn to explore other areas. Figure 4.13 illustrates the process of evaluating the fit values of a slime mold. Based on the principle above, the mathematical formula for updating a slime mold site is as follows.

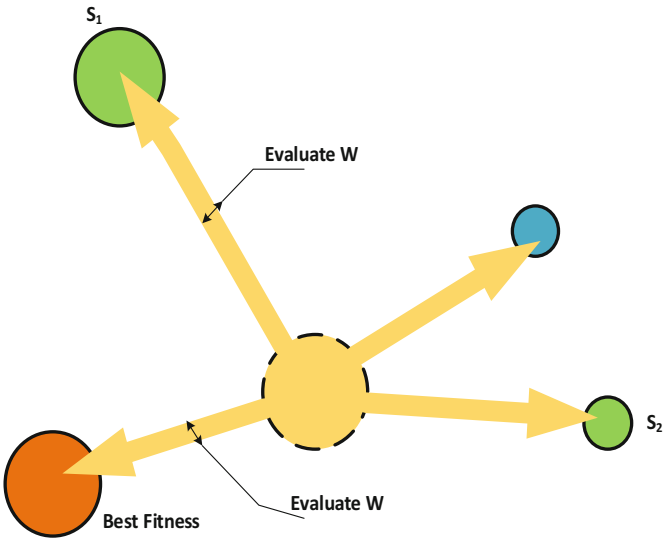

Fig. 4.13 Evaluation of fitness

$$X^* = \begin{cases} \text{rand} \cdot (\text{UB} - \text{LB}) + \text{LB}, \text{rand} < z \\ X_b(t) + vb \cdot (W \cdot X_{\text{A}}(t) - X_{\text{B}}(t)), r < p \\ vc \cdot X(t), r \geq p \end{cases} \tag{4.79}$$

where LB and UB are the bounds of the lower and upper control variables, and rand and r denote the random value in [0, 1].

4.8.3 Grabble Food

The value of vb randomly varied within range $[-a, a]$ and gradually reached zero as the iteration number increases. The value of vc varied between $[-1, 1]$ and reached zero eventually (Fig. 4.14).

Algorithm 1 Pseudo-code of SMA

Define the parameters of SMA the maximum number of iterations.

Generate initial positions of slime mold $X_i(i = 1, 2, \ldots, n)$;

While ($t \leq$ *Max _ iteraition*)

Evaluate the objective function of all slime mold.

Update bestFitness, X_b

Obtain the W using Eq. (4.77).

For *each search portion*

Update p, *vb*, *vc*;
Update locations by Eq. (4.79);

End For
$t = t + 1$;
End while
Return bestFitness, X_b;

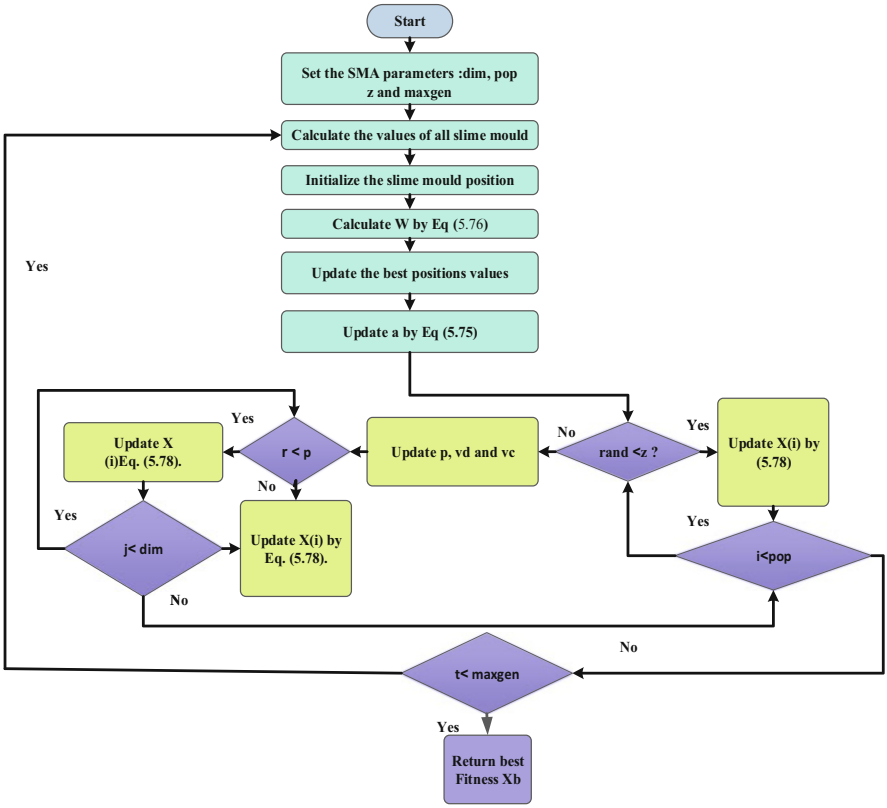

Fig. 4.14 Flowchart of SMA optimization technique