A brief description of the GBO algorithm

**1. Gradient-based optimizer**

**1.1. Gradient search rule (GSR)**

The proposed helps the GBO to account for the random behavior during the optimization process, promoting exploration and escaping local optima. The direction of movement (*DM*) is used to create a suitable local search tendency to promote the convergence speed of the GBO algorithm. Based on the GSR and DM, the following equation is used to update the position of current vector ().

(1)

In which

(2)

|  |  |
| --- | --- |
|  | (2-1) |
|  | (2-2) |

where and are 0.2 and 1.2, respectively, is the number of iterations, and is the total number of iterations. is a normally distributed random number, and is a small number within the range of [0, 0.1]. is given by:

|  |  |
| --- | --- |
|  | (3) |
|  | (3-1) |
|  | (3-2) |

where is a random number with *N* dimensions, ) are different integers randomly chosen from [1, *N*], is a step size, which is determined by and .

By replacing the position of the best vector () with the current vector () in Eq. (1), the new vector () can be generated as follows:

|  |  |  |
| --- | --- | --- |
|  | | (4) |
| in which | |  |
|  | (4-1) | |
|  | (4-2) | |

Based on the positions , , and the current position (, the new solution at the next iteration () can be defined as:

(5)

(5-1)

**1.2. Local escaping operator (LEO)**

The LEO is introduced to promote the efficiency of the proposed GBO algorithm for solving complex problems. The LEO generates a solution with a superior performance () by using several solutions, which include the best position (), the solutions and , two random solutions and , and a new randomly generated solution (). The solution is generated by the following scheme:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
| **else** | (6) |
|  |  |
|  |  |
| **End** |  |
| **End** |  |

where is a uniform random number in the range of [-1,1], is a random number from a normal distribution with mean of 0 and standard deviation of 1, is the probability, and , , and are three random numbers, which are defined as:

(6-1)

(6-2)

(6-3)

where is a random number in the range of [0, 1], and is a number in the range of [0, 1]. The above equations can be simplified:

(6-4)

(6-5)

(6-6)

where is a binary parameter with a value of 0 or 1. If parameter is less than 0.5, the value of is 1, otherwise, it is 0. To determine the solution in Eq. (6), the following scheme is suggested.

(6-7)

(6-8)

where is a new solution, is a randomly selected solution of the population (]), and is a random number in the range of [0, 1]. Eq. (6-7) can be simplified as:

(6-9)

where is a binary parameter with a value of 0 or 1. If is less than 0.5, the value of is 1, otherwise, it is 0. The pseudo code of the GBO algorithm is shown in Table 1.

|  |
| --- |
| **Table 1** Pseudo code of the GBO algorithm |
| **Step 1. Initialization** |
| Assign values for parameters,, and *M* |
| Generate an initial population |
| Evaluate the objective function value |
| Specify the best and worst solutions and |
| **Step 2. Main loop** |
| **While (***m<M***)** |
| **for** *n* = 1 : *N* |
| **for** *i* = 1 : *D* |
| Select randomly in the range of [1, *N*] |
| Calculate the position using Eq. 5 |
| **end for** |
| **Local escaping operator** |
| **if** |
| Calculate the position using Eq. 6 |
|  |
| **end** |
| Update the positions and |
| **end for** |
| m=m+1 |
| **end** |
| **Step 3.** return |