**Brief description of RUN optimization algorithm**

**1 Updating solutions**

The RUN algorithm uses a search mechanism (SM) based on the Runge Kutta method to update the position of current solution at each iteration, which is defined as,

|  |
| --- |
|  |
| **(exploration phase)** |
| (1) |
| **(exploitation phase)** |
|  |

where is an integer number, which is 1 or -1. is a random number in the range [0, 2]. is an adaptive factor. where is a random number. The formula of is defined in Appendix A.

The formula of is as follows:

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

where stands for the largest number of iterations. The formula of and are as follows:

(3)

(4)

where is a random number in the range of (0,1). is the best-so-far solution. is the best position obtained at each iteration.

**2. Enhanced solution quality (ESQ)**

In the RUN algorithm, enhanced solution quality (ESQ) is employed to increase the quality of solutions and avoid local optima in each iteration. The following scheme is executed to create the solution (­) by using the ESQ:

|  |  |
| --- | --- |
|  |  |
|  | (5) |
|  |
|  |
|  |
|  |
|  |  |

|  |  |
| --- | --- |
|  | (5-1) |
|  | (5-2) |
|  | (5-3) |

where is a random number in the range of [0, 1]. is a random number, which is equal to 5 in this study. is an integer number, which is 1, 0, or -1. is the best solution explored so far.

The solution calculated in this part () may not have better fitness than that of the current solution (i.e., ). To have another chance for creating a good solution, another new solution () is generated, which is defined as follows:

|  |  |
| --- | --- |
| **if** **rand<** |  |
|  | (6) |
| **end** |  |

where is a random number with a value of .

|  |
| --- |
| **Algorithm 1.**  The pseudo-code of RUN |
| **Stage 1. Initialization** |
| Initialize, |
| Generate the RUN population |
| Calculate the objective function of each member of population |
| Determine the solutions , , and |
| **Stage 2. RUN operators** |
| **for** *i= 1: Maxi* |
| **for** *n* = 1 : *N* |
| **for** *l* = 1 : *D* |
| Calculate position using Eq. 1 |
| **end for** |
| **Enhance the solution quality** |
| **if** |
| Calculate position using Eq. 5  **if**  **if** rand<  Calculate position using Eq. 6  **end**  **end** |
| **end**  Update positions and |
| **end for**  Update position |
| *i*=*i*+1 |
| **end** |
| **Stage 3.** return |

**Appendix A:**

The formula of is defined as,

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  | | | |
|  | |  | | | |
|  | |  | | | |
|  | |  | | | |
|  | | |  | |

where and are two random numbers in the range of [0, 1]. The formula of is defined as,

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

In this study, and are determined by the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |