|  |
| --- |
| **Algorithm 1:** *The**CO Algorithm* |
| 1: | Define the problem data, dimension (), and the initial population size () |
| 2: | Generate the initial population of cheetahs and evaluate the fitness of each cheetah |
| 3: | Initialize the population’s home, leader, and prey solutions |
| 4: |  |
| 5: |  |
| 6: |  desired maximum number of iterations |
| 7: |   |
| 8: | **while** **do** |
| 9: |  | Select members of cheetahs randomly |
| 10: |  | **for** each member **do**  |
| 11: |  | Define the neighbor agent of member  |
| 12: |  |  | **for** each arbitrary arrangement **do** |
| 13: |  |  |  | Calculate , , and  |
| 14: |  |  |  |  random numbers are chosen uniformly from 0 to 1 |
| 15: |  |  |  | **if**  **then**  |
| 16: |  |  |  |  |  a random number is chosen uniformly from 0 to 3 |
| 17: |  |  |  |  | **if**  **then**  |
| 18: |  |  |  |  |  | Calculate the new position of member in arrangement using Equation (1) // Search  |
| 19: |  |  |  |  | **else** |
| 20: |  |  |  |  |  | Calculate the new position of member in arrangement using Equation (3) // Attack |
| 21: |  |  |  |  | **end** |
| 22: |  |  |  | **else** |
| 23: |  |  |  |  | Calculate the new position of member in arrangement using Equation (2) // Sit-and-wait  |
| 24: |  |  |  | **end** |
| 25: |  |  | **end** |
| 26**:** |  |  | Update the solutions of member and the leader |
| 27: |  | **end** |
| 28: |  |  |
| 29: |  | **if** and the leader position doesn't change for a time, **then** // Leave the prey and go back home |
| 30: |  |  | Implement the leave the prey and go back home strategy and change the leader position |
| 31: |  |  | Substitute the position of member by the prey position |
| 32: |  |  |  |
| 33: |  | **end** |
| 34: |  |  |
| 35: |  | Update the prey (global best) solution |
| 36: | **end** |