**Puma Optimizer (PO): A Novel Metaheuristic Optimization Algorithm and its Application in Machine Learning**

|  |  |
| --- | --- |
| (1) |  |
| (2) |  |
| (3) |  |
| (4) |  |

|  |  |
| --- | --- |
| (5) |  |
| (6) |  |
| (7) |  |
| (8) |  |
| (9) |  |
| (10) |  |

|  |  |
| --- | --- |
| (11) |  |
| (12) |  |

|  |  |
| --- | --- |
|  | (13) |
|  | (14) |

|  |  |
| --- | --- |
|  | (15) |
|  | (16) |

|  |  |
| --- | --- |
|  | (17) |
|  | (18) |

|  |  |
| --- | --- |
|  | (19) |
|  | (20) |
| , | (21) |
|  | (22) |
|  | (23) |
|  | (24) |

`

|  |  |
| --- | --- |
| (25) | If > 0.5,  Otherwise, |
| (26) |  |

|  |  |
| --- | --- |
| (27) |  |
| (28) |  |
| (29) |  |
| (30) |  |

|  |  |
| --- | --- |
|  | (31) |

|  |
| --- |
|  |
| Figure (4): PO exploration procedure |

|  |
| --- |
| Algorithm 1 Pseudo-code of the exploration phase of PO |
| Sort ascending population  *for* (each Puma (*Xi*)) *do*  Select four solutions randomly  Calculate the new vector by Eq. (24).  Check the boundary of the new vector position  Update the current solution by Eq. (26).  Calculate the cost of the updated solution  *if* *NewXi* Cost < *Xi Cost*  *Xi* = *NewXi*  *else*  ;  *end*  *end* |

|  |  |
| --- | --- |
| (32) |  |

|  |  |
| --- | --- |
| (33) |  |

|  |  |
| --- | --- |
| (34) |  |
| (35) |  |

|  |  |
| --- | --- |
| (36) |  |
| (37) |  |
| (38) |  |

|  |
| --- |
|  |
| Figure (5): PO exploitation procedure |

|  |
| --- |
| Algorithm 2 Pseudo-code of exploitation phase of PO |
| *for* (each Puma (*Xi*)) *do*  Calculate *R, F1* and *F2* by Eqs.(34), (35) and (36) respectively  Produce *NewXi* by Eq. (32)  Calculate the cost of *NewXi*  *if* *NewXi* Cost < *Xi Cost*  *Xi* = *NewXi*  *end*  *end* |

|  |
| --- |
| Algorithm 3 Pseudo-code of PO |
| % PO setting  Inputs: The population size N and the maximum number of iterations and parameter settings  Outputs: Puma’s location and fitness potential  % initialization  Create a random population using *Xi (i = 1, 2, ..., N).* Calculate Puma’s fitness levels.  *for* *iter = 1 : 3*  Apply exploration phase (*Algorithm 1*)  Apply exploitation phase (*Algorithm 2*)  *End*  Apply Unexperienced Phase  *for* *iter = 4: Max iteration*  Apply experienced Phase  *if Score Explore > Score Exploit*  Apply exploration phase (*algorithm 1*)  *if* Exploration *NewBestXi* Cost < *Puma male Cost*  *Puma male* = *NewBestXi*  *end*  *else*  Apply exploitation phase (*algorithm 2*)  *if* Exploitation *NewBestXi* Cost < *Puma male Cost*  *Puma male* = *NewBestXi*  *end*  *end*  Update *, f1 , f2* and *f3*  Update *Score Explore* and *Score Exploit* by Eq. (17)  *end* |

|  |
| --- |
|  |
| Figure (6): PO optimization procedure |