GA-HW2 Simulated Annealing

求最小值, x 範圍 -1.5 ~ 1.5

實驗環境:

srand(114)

實驗次數:100次

OS:Windows

Point: 每間隔0.03灑一個點，包含-1.5&1.5

Steps: 從0.1開始漸減至0.01、0.001;當機率接受較差解時一次跳1

setprecision(10)

實驗結果:

|  |  |
| --- | --- |
| 計算次數平均 | 最佳解平均 |
| 98 | 0.0000000000 |

100次實驗詳細數據:

|  |  |  |  |
| --- | --- | --- | --- |
| 實驗# | 起點 | 計算次數 | 最佳解(f(x)) |
| 1 | -1.5 | 111 | (0.0000000000,0.0000000000) |
| 2 | -1.47 | 150 | (0.0000000000,0.0000000000) |
| 3 | -1.44 | 121 | (0.0000000000,0.0000000000) |
| 4 | -1.41 | 80 | (0.0000000000,0.0000000000) |
| 5 | -1.38 | 125 | (0.0000000000,0.0000000000) |
| 6 | -1.35 | 84 | (0.0000000000,0.0000000000) |
| 7 | -1.32 | 3 | (0.0000000000,0.0000000000) |
| 8 | -1.29 | 81 | (0.0000000000,0.0000000000) |
| 9 | -1.26 | 75 | (0.0000000000,0.0000000000) |
| 10 | -1.23 | 117 | (0.0000000000,0.0000000000) |
| 11 | -1.2 | 74 | (0.0000000000,0.0000000000) |
| 12 | -1.17 | 91 | (0.0000000000,0.0000000000) |
| 13 | -1.14 | 113 | (0.0000000000,0.0000000000) |
| 14 | -1.11 | 130 | (0.0000000000,0.0000000000) |
| 15 | -1.08 | 96 | (0.0000000000,0.0000000000) |
| 16 | -1.05 | 119 | (0.0000000000,0.0000000000) |
| 17 | -1.02 | 92 | (0.0000000000,0.0000000000) |
| 18 | -0.99 | 115 | (0.0000000000,0.0000000000) |
| 19 | -0.96 | 132 | (0.0000000000,0.0000000000) |
| 20 | -0.93 | 134 | (0.0000000000,0.0000000000) |
| 21 | -0.9 | 62 | (0.0000000000,0.0000000000) |
| 22 | -0.87 | 115 | (0.0000000000,0.0000000000) |
| 23 | -0.84 | 108 | (0.0000000000,0.0000000000) |
| 24 | -0.81 | 120 | (0.0000000000,0.0000000000) |
| 25 | -0.78 | 99 | (0.0000000000,0.0000000000) |
| 26 | -0.75 | 108 | (0.0000000000,0.0000000000) |
| 27 | -0.72 | 69 | (0.0000000000,0.0000000000) |
| 28 | -0.69 | 122 | (0.0000000000,0.0000000000) |
| 29 | -0.66 | 97 | (0.0000000000,0.0000000000) |
| 30 | -0.63 | 146 | (0.0000000000,0.0000000000) |
| 31 | -0.6 | 77 | (0.0000000000,0.0000000000) |
| 32 | -0.57 | 76 | (0.0000000000,0.0000000000) |
| 33 | -0.54 | 117 | (0.0000000000,0.0000000000) |
| 34 | -0.51 | 116 | (0.0000000000,0.0000000000) |
| 35 | -0.48 | 87 | (0.0000000000,0.0000000000) |
| 36 | -0.45 | 80 | (0.0000000000,0.0000000000) |
| 37 | -0.42 | 60 | (0.0000000000,0.0000000000) |
| 38 | -0.39 | 118 | (0.0000000000,0.0000000000) |
| 39 | -0.36 | 131 | (0.0000000000,0.0000000000) |
| 40 | -0.33 | 129 | (0.0000000000,0.0000000000) |
| 41 | -0.3 | 113 | (0.0000000000,0.0000000000) |
| 42 | -0.27 | 9 | (0.0000000000,0.0000000000) |
| 43 | -0.24 | 72 | (0.0000000000,0.0000000000) |
| 44 | -0.21 | 88 | (0.0000000000,0.0000000000) |
| 45 | -0.18 | 92 | (0.0000000000,0.0000000000) |
| 46 | -0.15 | 122 | (0.0000000000,0.0000000000) |
| 47 | -0.12 | 62 | (0.0000000000,0.0000000000) |
| 48 | -0.09 | 115 | (0.0000000000,0.0000000000) |
| 49 | -0.06 | 115 | (0.0000000000,0.0000000000) |
| 50 | -0.03 | 100 | (0.0000000000,0.0000000000) |
| 51 | 0 | 0 | (0.0000000000,0.0000000000) |
| 52 | 0.03 | 77 | (0.0000000000,0.0000000000) |
| 53 | 0.06 | 117 | (0.0000000000,0.0000000000) |
| 54 | 0.09 | 114 | (0.0000000000,0.0000000000) |
| 55 | 0.12 | 91 | (0.0000000000,0.0000000000) |
| 56 | 0.15 | 98 | (0.0000000000,0.0000000000) |
| 57 | 0.18 | 56 | (0.0000000000,0.0000000000) |
| 58 | 0.21 | 113 | (0.0000000000,0.0000000000) |
| 59 | 0.24 | 118 | (0.0000000000,0.0000000000) |
| 60 | 0.27 | 140 | (0.0000000000,0.0000000000) |
| 61 | 0.3 | 92 | (0.0000000000,0.0000000000) |
| 62 | 0.33 | 127 | (0.0000000000,0.0000000000) |
| 63 | 0.36 | 124 | (0.0000000000,0.0000000000) |
| 64 | 0.39 | 109 | (0.0000000000,0.0000000000) |
| 65 | 0.42 | 125 | (0.0000000000,0.0000000000) |
| 66 | 0.45 | 134 | (0.0000000000,0.0000000000) |
| 67 | 0.48 | 98 | (0.0000000000,0.0000000000) |
| 68 | 0.51 | 106 | (0.0000000000,0.0000000000) |
| 69 | 0.54 | 109 | (0.0000000000,0.0000000000) |
| 70 | 0.57 | 117 | (0.0000000000,0.0000000000) |
| 71 | 0.6 | 122 | (0.0000000000,0.0000000000) |
| 72 | 0.63 | 10 | (0.0000000000,0.0000000000) |
| 73 | 0.66 | 58 | (0.0000000000,0.0000000000) |
| 74 | 0.69 | 91 | (0.0000000000,0.0000000000) |
| 75 | 0.72 | 71 | (0.0000000000,0.0000000000) |
| 76 | 0.75 | 134 | (0.0000000000,0.0000000000) |
| 77 | 0.78 | 72 | (0.0000000000,0.0000000000) |
| 78 | 0.81 | 75 | (0.0000000000,0.0000000000) |
| 79 | 0.84 | 92 | (0.0000000000,0.0000000000) |
| 80 | 0.87 | 118 | (0.0000000000,0.0000000000) |
| 81 | 0.9 | 68 | (0.0000000000,0.0000000000) |
| 82 | 0.93 | 109 | (0.0000000000,0.0000000000) |
| 83 | 0.96 | 105 | (0.0000000000,0.0000000000) |
| 84 | 0.99 | 115 | (0.0000000000,0.0000000000) |
| 85 | 1.02 | 96 | (0.0000000000,0.0000000000) |
| 86 | 1.05 | 98 | (0.0000000000,0.0000000000) |
| 87 | 1.08 | 51 | (0.0000000000,0.0000000000) |
| 88 | 1.11 | 103 | (0.0000000000,0.0000000000) |
| 89 | 1.14 | 89 | (0.0000000000,0.0000000000) |
| 90 | 1.17 | 117 | (0.0000000000,0.0000000000) |
| 91 | 1.2 | 65 | (0.0000000000,0.0000000000) |
| 92 | 1.23 | 58 | (0.0000000000,0.0000000000) |
| 93 | 1.26 | 121 | (0.0000000000,0.0000000000) |
| 94 | 1.29 | 116 | (0.0000000000,0.0000000000) |
| 95 | 1.32 | 89 | (0.0000000000,0.0000000000) |
| 96 | 1.35 | 121 | (0.0000000000,0.0000000000) |
| 97 | 1.38 | 90 | (0.0000000000,0.0000000000) |
| 98 | 1.41 | 125 | (0.0000000000,0.0000000000) |
| 99 | 1.44 | 128 | (0.0000000000,0.0000000000) |
| 100 | 1.47 | 92 | (0.0000000000,0.0000000000) |

程式說明:

宣告變數:

    int EXP\_TIME = 100; //實驗次數

double PI = 2 \* acos(0.0); //

double step[] = {0.1, 0.01, 0.001}; //步數

    int stepsize = sizeof(step) / sizeof(step[0]); //step陣列大小

    double p = -1.5;       //撒點起點

    double p\_delta = 0.03; //每隔0.03撒一個點

    int all\_count = 0; //所有次數相加

    double all\_b = 0; //所有最佳解相加

    double x\_min = -1.5; //x的最小值

    double x\_max = 1.5; //x的最大值

double T = 3000.0; //初始溫度

    double af = 0.8;   //退火率

    int count = 0; //每次實驗的計算次數

    double x = p; //目前在哪

    double r;     //隨機0-1

設定終止條件:

當溫度小於0.0001

while (T > 0.0001)

機率接受較差解:

r為隨機亂數0~1，當r < exp(-(f(n+1)-f(n))/T) 時，會往較差解靠近，然後為了讓他跳比較遠我讓他一次跳1(嘗試過0.8、0.75、0.5)發現1為較好選擇

r = ((double)rand() / RAND\_MAX);

     if (r < exp(-(f(neighbor\_right, 1) - f(x, 1)) / T))

      {

       x = neighbor\_right;

       neighbor\_right = x + 1;

       neighbor\_left = x - 1;

       }

      else if (r < exp(-(f(neighbor\_left, 1) - f(x, 1)) / T))

       {

        x = neighbor\_left;

        neighbor\_right = x + 1;

        neighbor\_left = x - 1;

       }

f:

double f(double x, int d)

{

    double temp = 0;

    for (int i = 1; i <= d; i++)

    {

        temp += x \* x - 10 \* cos(2 \* PI \* x);

    }

    return 10 \* d + temp;

}