**使用智能优化算法求解多极小函数**

目录

[【方案一】SA模拟退火算法求解函数极小值](#_Toc25875_WPSOffice_Level1) [2](#_Toc25875_WPSOffice_Level1)

[1. 优化函数](#_Toc18681_WPSOffice_Level1) [2](#_Toc18681_WPSOffice_Level1)

[2. 算法设计](#_Toc26667_WPSOffice_Level1) [2](#_Toc26667_WPSOffice_Level1)

[3. 实验结果](#_Toc9875_WPSOffice_Level1) [3](#_Toc9875_WPSOffice_Level1)

[4. 典型变化曲线：](#_Toc7253_WPSOffice_Level1) [4](#_Toc7253_WPSOffice_Level1)

[5. 实验体会](#_Toc22862_WPSOffice_Level1) [5](#_Toc22862_WPSOffice_Level1)

[6. 算法的特点](#_Toc29714_WPSOffice_Level1) [5](#_Toc29714_WPSOffice_Level1)

[【方案二】使用遗传算法找最优解](#_Toc26985_WPSOffice_Level1) [6](#_Toc26985_WPSOffice_Level1)

[1. 函数：](#_Toc18540_WPSOffice_Level1) [6](#_Toc18540_WPSOffice_Level1)

[2. 算法：遗传算法](#_Toc32385_WPSOffice_Level1) [6](#_Toc32385_WPSOffice_Level1)

[3. 实验结果](#_Toc29273_WPSOffice_Level1) [6](#_Toc29273_WPSOffice_Level1)

[4. 实验过程当中最后化结果的变化曲线](#_Toc26166_WPSOffice_Level1) [7](#_Toc26166_WPSOffice_Level1)

[5. 实验体会](#_Toc10077_WPSOffice_Level1) [8](#_Toc10077_WPSOffice_Level1)

[6. 算法特点](#_Toc11487_WPSOffice_Level1) [9](#_Toc11487_WPSOffice_Level1)

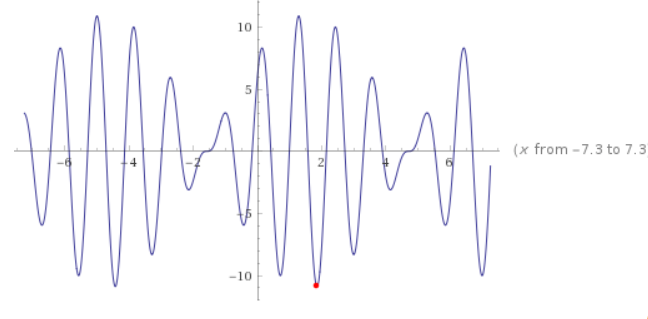
**【方案一】SA模拟退火算法求解函数极小值**

1. **优化函数**

minimize：

解析解：





（红点为真实的最小值点）

1. **算法设计**

使用基本的模拟退火算法。搜索区间：0—2π。

【算法步骤】

【1】给定初始温度，随机产生状态

【2】循环

【2.1】循环

【2.1.1】产生新的状态（在算法实现当中是随机产生）

【2.1.2】按概率选取该新解

【2.1.3】直到满足抽样稳定准则

【2.2】退温

【3】直到算法满足终止准则

【4】打印输出结果

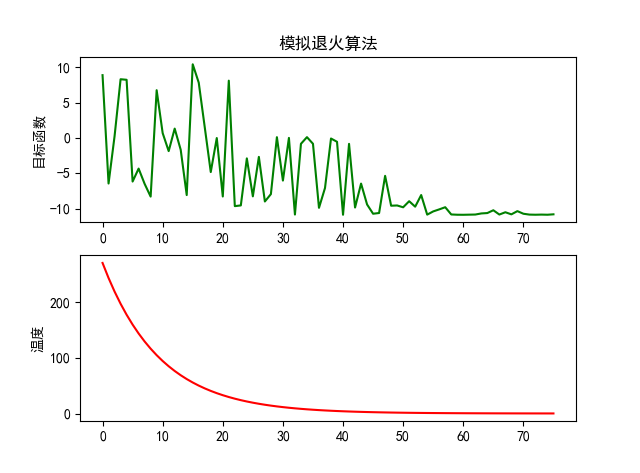
1. **实验结果**

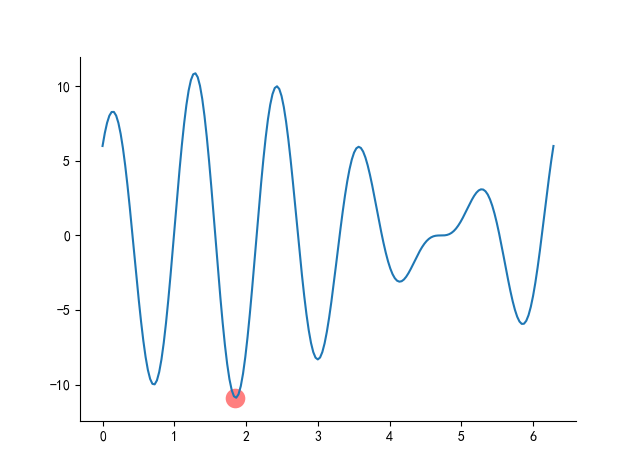
|  |  |  |  |
| --- | --- | --- | --- |
| 实验 | 优化结果 | 实验 | 优化结果 |
| 1 | -10.88358490738886 | 11 | -10.79859651531527 |
| 2 | -10.8637555375710 | 12 | -10.81687203491084 |
| 3 | -10.82359755974696 | 13 | -10.85412946665669 |
| 4 | -10.87493054370122 | 14 | -10.79446508068858 |
| 5 | -10.69066054328291 | 15 | -10.88182081113809 |
| 6 | -10.85044565593578 | 16 | -10.76647507568300 |
| 7 | -10.7949153967929 | 17 | -10.88624197476412 |
| 8 | -10.88607775803634 | 18 | -10.80057041809656 |
| 9 | -10.80170972440511 | 19 | -10.88803324365304 |
| 10 | -10.8871908978588 | 20 | -10.85954864725442 |

**统计结果：**

|  |  |
| --- | --- |
| 平均性能 | -10.835181089644037 |
| 最佳性能 | -10.888033243653041 |
| 最差性能 | -10.690660543282913 |
| 方差 | 0.002590423903365339 |

1. **典型变化曲线：**





1. **实验体会**

模拟退火算法作为一种经典的随机寻优算法，有其独特的好处，对于一般的优化问题，其都能得到比较好的优化结果。同时在阅读了相关文献之后发现，模拟退火算法有较强的通用性，在多个领域当中都有较好的表现。因此，复杂的算法并不一定是好的，相反，一些简单的算法，稍加变形，将可以在多个问题上表现出较好的结果。

1. **算法的特点**

通过实验可以发现，模拟退火算法计算过程比较简单，且鲁棒性强，多次实验都能达到较好的结果，且结果方差小，统计结果都有较好的表现。同时也发现了模拟退火算法的一些缺点，如果温度下降较快，很容易得到的不是全局最优解。为了得到全局最优解，就得使温度下降的足够慢，这样导致执行时间比较长。

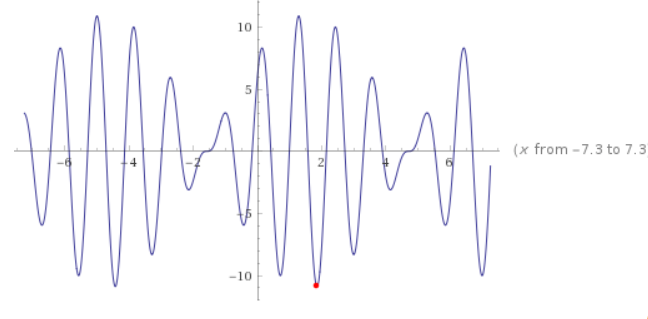
**【方案二】使用遗传算法找最优解**

1. **函数：**

优化函数：minimize：

解析解：





1. **算法：遗传算法**

【1】随机产生初始种群，直接用函数作为适配值函数

【2】判断算法是否满足收敛准则

【3】按照适配值函数的结果以一定的方式执行复制操作

【4】按照交叉概率执行交叉

【5】按照变异概率执行变异

【6】返回第【2】步

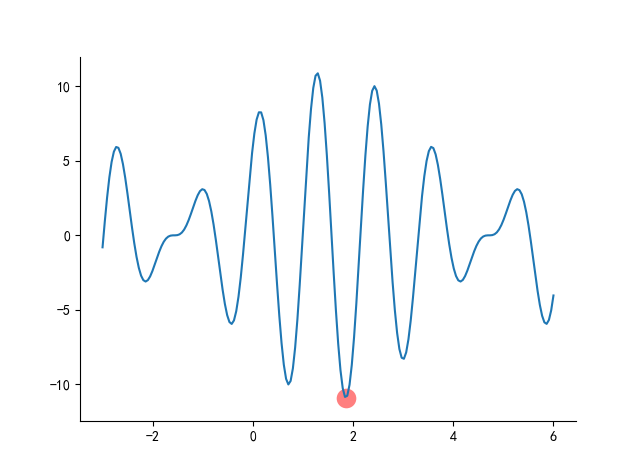
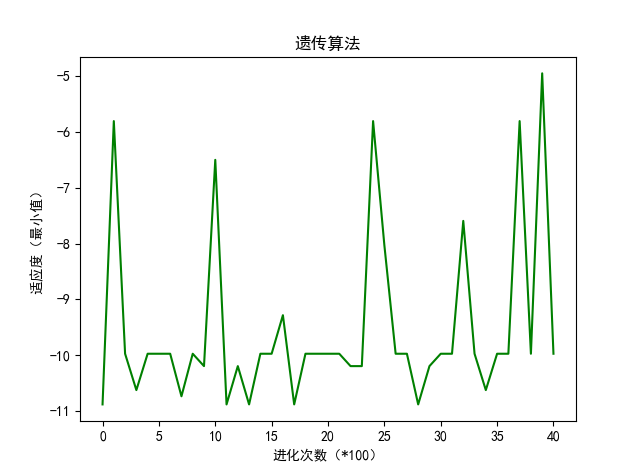
1. **实验结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 实验 | 优化结果 | 实验 | 优化结果 |
| 1 | -10.197681399252223 | 11 | -10.884220133355495 |
| 2 | -9.975894292041833 | 12 | -9.975894292041833 |
| 3 | -9.975894292041833 | 13 | -10.197681399252223 |
| 4 | -9.975894292041833 | 14 | -9.975894292041833 |
| 5 | -10.884220133355495 | 15 | -5.952244933859539 |
| 6 | -9.975894292041833 | 16 | -9.975894292041833 |
| 7 | -10.627440360235237 | 17 | -5.80543856968869 |
| 8 | -10.884220133355495 | 18 | -9.975894292041833 |
| 9 | -9.975894292041833 | 19 | -10.627440360235237 |
| 10 | -10.627440360235237 | 20 | -9.975894292041833 |

**统计结果：**

|  |  |
| --- | --- |
| 平均性能 | -9.82234853516216 |
| 最佳性能 | -10.884220133355495 |
| 最差性能 | -5.80543856968869 |
| 方差 | 1.8479006240686353 |

1. **实验过程当中最后化结果的变化曲线**



1. **实验体会**

使用遗传算法在进行对极小函数时，对于问题的编码和解码的过程比较麻烦，实现起来也比上面的模拟退缩要难。因此，就目前的体验来讲，对于一般的比较简单的组合优化问题，使用原理简单、容易实现的算法就可以得到比较不错的解。

1. **算法特点**

通过实验发现，算法能在较短的时间当中得到最优值，且种群的概念使得算法有并行的模式，多的个体同时搜索最优值。但算法的实现比较麻烦，设计到问题的编码和解码，且参数较多，不同的参数选择选择也没有理论的好坏，只能通过多次尝试。同时由于遗传算法的迭代眼里，使得初始解对最终解的影响比较大。

特别的，由于本次实验所采用的函数比较简单，对于所使用的函数比较简单，如果参数选择不太合适，有时得到的解会在别的局部极小，比如：

