# 计算智能——模拟退火算法

作者 柯良军

西安交通大学 电信学院

October 20, 2015

# 目录

# 特点

研究表明，模拟退火算法具有以下优势：

- 强全局搜索能力：在搜索最优解时，不但往好的方向搜索，也往差的方向搜索，因而可以有效地跳出局部最优。
- 强鲁棒性：算法达到全局最优不依赖初始值。
- 渐进收敛性：理论上已经证明在一定条件下以概率1收敛于全局最优解（或目标值最小的状态）。

# 特点



研究表明，模拟退火算法具有以下优势：

- 强全局搜索能力：在搜索最优解时，不但往好的方向搜索，也往差的方向搜索，因而可以有效地跳出局部最优。
- 强鲁棒性：算法达到全局最优不依赖初始值。
- 渐进收敛性：理论上已经证明在一定条件下以概率1收敛于全局最优解（或目标值最小的状态）。

然而，在用模拟退火算法也求解问题时，一些算法要素不容易设计。例如，降温过程要足够的缓慢，以使得在每一个温度下达到热平衡，这样做的好处是可返回一个高质量的解，但是计算时间开销大，算法收敛过慢。如果降温速度过快，很可能产生一个很差的解。

# 有记忆的模拟退火算法



模拟退火算法是一种全局搜索算法，可以有效的跳出局部最优，在搜索过程中根据Metropolis规则不但接受改进解而且以一定概率接受使目标函数值变差的解。在实际工程应用中，许多问题的目标函数是多峰，它们存在多个极值点。在用模拟退火算法求解时，往往会在多个局部极小区域轮流搜索，从而导致算法性能不佳。一个解决途径是增加一个变量用于记忆搜索过程中达到的最好结果。具体而言，在每次接受使目标函数值下降的解时，将其与以前搜索到的最好解比较，取较优者更新最好解，最终输出最好解即为能搜索到的最好解。

# 带有单调升温的模拟退火算法



模拟退火算法在初始温度足够高时，以接近1的概率接受所有新解进行全局搜索，有很强的跳出局部最优的能力；温度降到足够低时，只接受优于当前解的新解，进行局部搜索。如果算法进行局部搜索陷入局部最优时，将很难跳出局部最优或者要花相当长的时间才能跳出局部最优。因此当算法陷入局部最优后，人为地升高温度，使算法能快速地跳出局部最优。

算法陷入局部最优有两个重要特征：一是在最近的$k$ 次搜索中最优解没有更新；另一个是在最近的$k$ 次搜索中有几个目标函数相同的解被重复接受。我们可以结合实际的问题设定阈值$k$，如果出现上述两个特征，则说明算法已经陷入局部最优，而后应该升温使算法尽快跳出局部最优。升温是为了提高较差解的接受概率，若升温幅度过小，达不到效果；若升温幅度过大，搜索又会进入漫长的全局搜索。根据实际经验，一般将温度升高到使接受概率为$60\% \sim 70\%$，这样可以保证搜索尽快跳出局部最优，也可避免进入漫长的全局搜索。

# 并行模拟退火算法



模拟退火算法是一种随机近似算法，接受的新解仅和前解有关，是一种串行的的过程，求解的质量有赖于大量实验。随着问题规模的增大和对解的质量要求的提高，求解时间也随之增高。虽然选取合适的初始温度和冷却进度表能使算法得到满意的结果，但是并不能从根本上提高算法的效率。因此将算法并行实现可以提高算法的效率。

# 并行模拟退火算法

根据算法和组合优化问题的特点，可以得到下面几种并行策略：

（1）操作并行：可以将产生解，计算目标函数差值和根据Metropolis规则三个步骤分别由三个处理机来并行执行。操作并行中可并行的步骤数受具体问题的限制，整个算法只能同步并行，但不同的步骤可分别并行。

（2）试验并行：由n台处理机同时从n个初值分别开始试验。在每次内循环中选取所有处理机接受新解中的最优一个作为当前最优解，并进行下次循环。这种策略很好的利用了并行计算的优越性，可以得到较为满意的解。

# 并行模拟退火算法



（3）解空间分裂并行：将解空间分解成n个子空间，并且有n台处理机分别在这些子空间上求解，输出所有子空间最优解中最好的解。在解空间分裂并行中，每个处理机都是完全独立的，异步并行的。当目标函数有多个极值时，合理的解空间分解可以避免算法陷入局部最优或者陷入局部最优的次数大幅度减少。

（4）混乱松弛并行：各个处理机使用同一个公共的当前解，同时进行试验，每个处理机进行完试验可以直接对公共的当前解进行修改和使用。这样当前解的更新将是由各个处理机竞争的，因此是混乱的。但随着温度$T$ 的减少，接受概率越来越小，解也逐渐趋于有序。混乱松弛法采用“不怕出错”的思想，是一种高效的异步并行算法，但其渐进收敛性尚待进一步研究。

# 连续优化



下面采用模拟退火算法来求解遗传算法章节中优化问题，问题描述如下：

$$\min f(x) = 4x^3 + 3x^2 - 6x + 1, x \in [-1, 1] \tag{1}$$

设定求解的精度为小数点后2位，已知该问题的最小解是$x = 0.5$，其对应的最优目标函数值为-0.75。下面就模拟退火算法求解这个简单问题。

# 连续优化



解：针对这个问题采用自然数编码，例如$i=75$ 表示$i$ 实际值0.75 。具体步骤如下：

# 连续优化

解：针对这个问题采用自然数编码，例如$i = 75$ 表示$i$ 实际值0.75 。具体步骤如下：

1. 初始化：从解空间中随机生成一个解作为初始解，假设初始解$i = 75$，初始温度$T = 10$，
最大降温次数$k_{max} = 50$，设置最大内循环次数$n_{max} = 4$，令降温次数$k = 0$。

# 连续优化

2. 产生新解：新解$j$ 可以按照函数$j = i + randint([-m, m])$产生，其中$randint([-m, m])$ 函数产生一个随机整数$\xi \in [-m, m]$，$m$ 为邻域的半径，假设等于8。

检验新解是否合法，即新解属于解空间，如果新解合法，则计算目标函数差$\Delta f = f(j) - f(i)$；否则重复步骤2。

3. 接受新解：如果$\Delta f < 0$ 成立无条件接受新解$j$，例如新解$j = 70$，
那么$\Delta f = f(j) - f(i) = -0.3580 - (-0.1250) = -0.2330 < 0$，
更新当前解$i = j$，$f(i) = f(j)$。若$\Delta f > 0$ ,判断$exp(-\frac{\Delta f}{T}) > random(0, 1)$ 是否成立，
若成立接受；否则不接受。
例如新解$j = 80$，那么$\Delta f = f(j) - f(i) = 0.1680 - (-0.1250) = 0.2930 > 0$，
则接受概率$p = exp(-\frac{\Delta f}{T}) = e^{-\frac{0.2930}{10}} = 0.9711$，
假设$random(0, 1) < p$，则接受新解更新当前解；假设$random(0, 1) > p$，则不接受。

# 连续优化

4. 内循环终止条件：若内循环次数$n = n + 1 > n_{max}$ 成立，则转第5步；否则转步骤2。

# 连续优化



4. 内循环终止条件：若内循环次数$n = n + 1 > n_{max}$ 成立，则转第5步；否则转步骤2。

5. 降温：若降温次数$k = k + 1 > k_{max}$ 成立，则算法结束；否则更新温度$T = 0.95T$ ，转步骤2。