
算法设计

模拟退火算法是一种模拟自然界物理过程的算法。通过模仿固体从高温徐徐冷却，内部分子从无序逐渐变成有序，内能减为最小的过程，使目标函数的值也达到最小。也就是说，我们可以设置一个变量来表示温度，而温度随循环不断减小，便是降温的过程。大量研究和实践保证了算法的较优性质，在收敛时有可观的效果。对于流水车间调度问题，本文针对目标函数特殊设计了计算的方法，利用了工件加工的有序性，前驱性，将约束条件转化为了 DAG 图上的边，通过遍历图的方式即可得出每个状态对应目标函数的值。

具体而言，我们可以按照以下几步实现模拟退火算法。

步骤一：设置一个初始温度 T ，一个结束温度零温 T_0 ，还有指数衰减率 $rate$

步骤二：如果初始温度 $T \leq$ 零温 T_0 ，则算法结束，将目前最优解作为答案

步骤三：随机产生一个当前状态邻域内的新状态，并用 DAG 评价函数估值

步骤四：如果新状态更优则接受；否则取两个状态估值的差 $\delta = \text{new_value} - \text{old_value}$ ，以指数分布的概率接受新状态

步骤五：温度下降，转到步骤二

下面给出算法伪代码

function Simulated-Annealing(problem, schedule) returns a solution state

inputs: problem, a problem

T , highest temperature

T_0 , lowest temperature

$rate$, temperature's down rate

local variables: $current$, current state

$next$, next state

$current \leftarrow$ Random-initialization

while $T > T_0$ begin

$next \leftarrow$ a randomly selected successor of $current$

DAG-evaluate $next$

$\delta \leftarrow \text{new-value} - \text{old-value}$

if $\delta > 0$ then $current \leftarrow next$

else $current \leftarrow next$ only with probability $\exp(\delta/T)$

$T \leftarrow T * (1 - rate)$

end

分析算法时间复杂度。可以看到算法的主体是一个循环，设循环执行了 k 次，则有 $T_0 = T(1 - rate)^k$ ，可得 $k = \log_{(1-rate)} \frac{T_0}{T}$ ，同时对于 DAG-evaluate 函数，建图复杂度 $O(NM)$ ，进行拓扑序动态规划复杂度 $O(N + M)$ ，

故总体复杂度为 $O\left(NM \log_{(1-rate)} \frac{T_0}{T}\right)$ ，远低于寻找准确最优解所需的指数级复杂度。

分析空间复杂度。我们仅在 DAG-evaluate 函数中和状态的存储中使用到了一部分内存，并且复杂度均为 $O(NM)$ ，故在空间复杂度上也是可以接受的。