问题描述：

旅行商问题（TSP）是一个组合优化问题，具有 NPC 计算复杂性。

主要内容是一位商人要拜访 N 座城市，但每个城市只能访问一次，最后要回到出发的城市，求解最短访问路径。

算法选择：

1. 穷举法：最简单最暴力的方法就是穷举法，最坏情况下时间复杂度为 O(n!)，存在维度爆炸问题，对于规模略大一点的问题基本无法实现。
2. 动态规划：动态规划可以将时间复杂度降为 O(2^n)，同样具有维度爆炸的问题，只是较穷举法有改善，但对于大规模问题仍束手无策。
3. 启发式算法：常见的启发式算法有蚁群（鱼群）算法、模拟退火、遗传算法、粒子群算法、天牛须搜索算法等。每个算法各有优缺点，但都跟初始参数尤其是随机种子的选择有很大关系，都需要重新搜索多次才能得到最优结果。在本文中，我将选择模拟退火作为 TSP 问题的求解算法，并做了一些消融实验。
4. Hopfield 神经网络：这个方法往往不稳定，且结果较差。

算法详解：

在介绍模拟退火算法之前，有必要先讲一下爬山法，爬山法可以认为是模拟退火的纯贪婪部分。

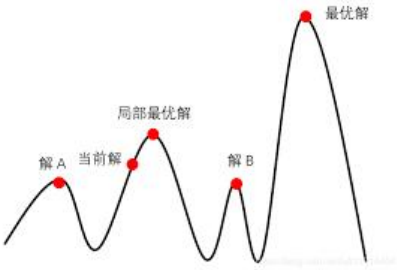


图 1 爬山法

爬山法是从当前解的临近解空间选择一个最优解作为当前解，直到局部最优解。这种纯贪婪的算法往往陷入局部最优，常见的解决方法是随机重启爬山算法。如图 1 所示，爬山法往往使得当前解陷入局部最优解。

模拟退火是模拟了物理上物体降温的过程。物理学家发现，当对物体充分加热，再徐徐降温会使得物体内部粒子排列逐渐趋向有序，最后形成晶体，而晶体也是物体的最低能量状态（如图 2 所示）；若快速降温，物理内部粒子则很难形成晶体状态。

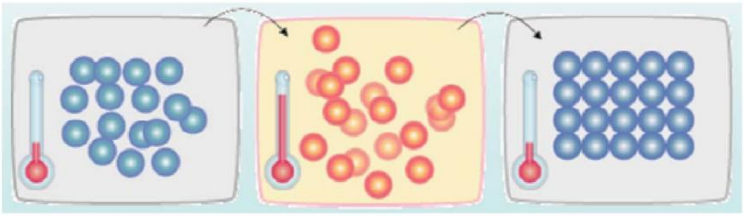


图 2 模拟退火物理模型示意图

模拟退火就是模仿了这一思想，从较高温度开始，随着温度下降，以一定概率接受比当前解较差的解，使得算法有机会跳出局部最优解，而得到全局最优解。

模拟退火主要包括两部分：Metropolis 算法和退火算法。Metropolis 算法是 Metropoli1953 年提出的重要性采样方法，即依概率来接受新状态，而不是完全确定的规则。

（1）Metropolis 算法：

假设系统状态为 x(n)，按照某一策略，状态变为 x(n+1)，相应地，系统能量有 E(n)变为 E(n+1)，那么允许系统由状态 x(n)变为状态 x(n+1)的概率是：

$$P=\begin{cases}1, E(n+1)<E(n)\\ e^{-\frac{F(n+1)-\mathrm{Er}(n)}{T}}, E(n+1)\geq E(n)\end{cases}$$

当下一个状态 x(n+1)对应的系统能量 E(n+1)变小时，系统转变为下一个状态 x(n+1)的概率为 1，而 E(n+1)变大时，则采取 $e^{-\frac{B(n+1)-R(n)}{T}}$ 的概率转移到下一个状态 x(n+1)，使得算法有机会跳出局部最优值。

（2）退火算法：

从 Metropolis 算法公式我们可以看到，可以调节的参数是 T，T 越大，转移到下一个状态的概率也就越大。所以为了在初期算法能够尽量遍历所有状态，T 的初始值往往较大，随着算法不断迭代，T 值逐步减小。T 值最简单的下降方式就是指数式下降：

$$T(n+1)=\lambda T(n), n=1,2,3,\ldots$$

还有另两种下降方式，会使得退火速度更快些：

$$T(n)=\frac{T(0)}{\log(1+t)}$$

$$T(n)=\frac{T(0)}{1+t}$$

对于终止温度一般设置较小，如 0.001，使得最终状态可以稳定。

当算法到达终止温度或者达到用户设定阈值或者没有可更新的新状态时，退火完成。

算法基本步骤：

1. 初始化温度 T，初始解状态 x(n)，每个温度 t 下的迭代次数 L；
2. 当 k = 1，2，……，L 时，进行 3~6；
3. 对当前解进行变换得到新解 x(n+1)；
4. 计算增量 ΔE=E(n+1)-E(n)，其中 C(S)为评价函数；
5. 若 ΔE<0,接受 x(n+1)作为新的当前解，否则以概率 exp(-ΔE/(T))接受 x(n+1)作为新的当前解
6. 如果满足终止条件则输出当前解作为最优解，结束程序；
7. 减小 T，转到第 2 步，直到 T 小于初始设定的阈值。

实验结果：

1. 实验设置

实验平台：window10，Core i7 8th 笔记本电脑，本实验只是用单核 CPU。

默认参数（若之后未更改相应参数，则参数值等于默认参数值）：

初始温度 T=200，温度衰减率 gamma=0.95，终止温度 final_T=0.001，迭代次数 L=120，初始解随机生成。

2. 消融实验

2.1 初始温度 T

	10	30	50	100	150	200	240	300
时间 T	2.01	2.01	2.83	2.537	2.688	2.71	2.67	2.75
路径长度 Dis	722.8	730.8	730.0	746.0	726.4	712.2	731.4	728.8

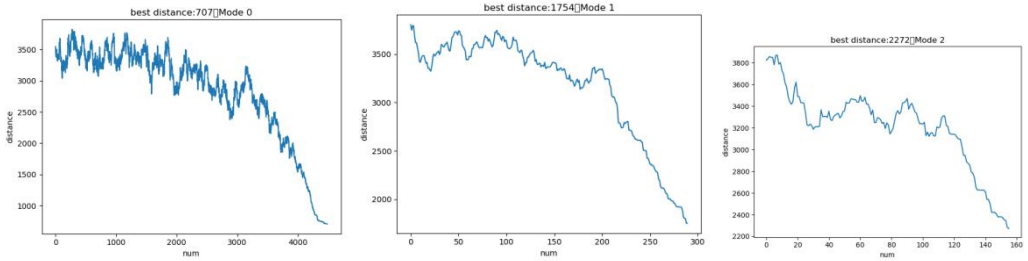
注：同初始温度都测试 5 个不同随机种子后取平均，取平均后距离 distance 保留到小数点一位。

初始温度越高所用时间越长，但时间增长并不是特别明显。针对本实验来说，随着初始温度的提高，最终结果并没有并没有什么提高。

2.2 温度衰减方法

	Mode 0	Mode 1	Mode 2
时间 T	2.68	0.097	0.046
路径长度 Dis	726.2	1721.8	2245.2

注：每种不同温度衰减方法都测试 5 个不同随机种子后取平均，取平均后距离 distance 保留到小数点一位。



每种模式分别对应 $T(n+1)=\lambda T(n), n=1,2,3,\ldots$，$T(n)=\frac{T(0)}{\log(1+t)}$，$T(n)=\frac{T(0)}{1+t}$。可以看出指数方式的温度衰减方式运行时间最长，另两种衰减更快。从性能上，衰减越快性能越差，这是衰减太快算法没有进行充分搜索的原因。

2.3 温度衰减率 gamma

	0.1	0.3	0.5	0.8	0.9	0.95	0.98	0.99
时间 T	0.086	0.15	0.22	0.63	1.29	2.88	7.56	15.25
路径长度 Dis	1797.8	1419.0	1261.0	892.0	782.2	725.2	712.0	698.2

注：每种不同温度衰减率都测试 5 个不同随机种子后取平均，取平均后距离 distance 保留到小数点一位。

温度衰减率对时间影响很大，gamma 在 0.1~0.8 之间时间变化相对较缓，0.9~0.99 之间时间提高了一个数量级。所以，gamma 越高搜索时间越长，效果也越好。

2.4 迭代次数 L

	10	50	100	150	200
时间 T	0.27	1.30	2.31	3.94	5.32
路径长度 Dis	1167.6	802.2	747.4	719.2	716.8

注：每种不同内部循环迭代次数都测试 5 个不同随机种子后取平均，取平均后距离 distance 保留到小数点一位。

内部循环迭代次数对时间和最终性能的影响也很大，迭代次数越大，时间越长，性能越好，可以认为是充分搜索的结果。

2.5 状态转移模式

	cross	reverse
时间 T	2.447	2.717
路径长度 Dis	982.4	727.8

注：每种不同状态转移模式都测试 10 个不同随机种子后取平均，取平均后距离 distance 保留到小数点一位。

常见获取新解的方式是 cross 方式，即随机交换序列的两个位置，但经过长时间不同随机种子的测试，效果均不理想。采用 2-opt（reverse）算法后性能得到显著提高，2-opt 算法是 Glover 在 1986 年提出的禁忌搜索算法，常被用来解决 TSP 问题，主要思想是序列中随机选择两个位置，把位置之间的片段反向来获取新解。直觉上，cross 算法步长小，更难走出局部最优，reverse 算法步长可变，所以具有明显优势。

2.6 随机种子

	0	100	200	300	400	500	600	700	800	900	999
时间 T	2.34	2.19	2.40	2.16	2.23	2.35	3.51	3.66	3.67	3.47	3.43
路径长度 Dis	763	732	732	706	702	694	706	730	676	712	704

注：测试 0-999 的随机种子，这里平均选取 10 个作为参考，更多信息请参考 code/experiment/2.6_seed.txt。

可以看出所选算法对随机种子具有一定的鲁棒性，随机种子不同，算法性能波动并不明显。但要找到最优解，还是需要对随机种子进行搜索，实验结果显示随机种子为 800 时，性能最好为 676。