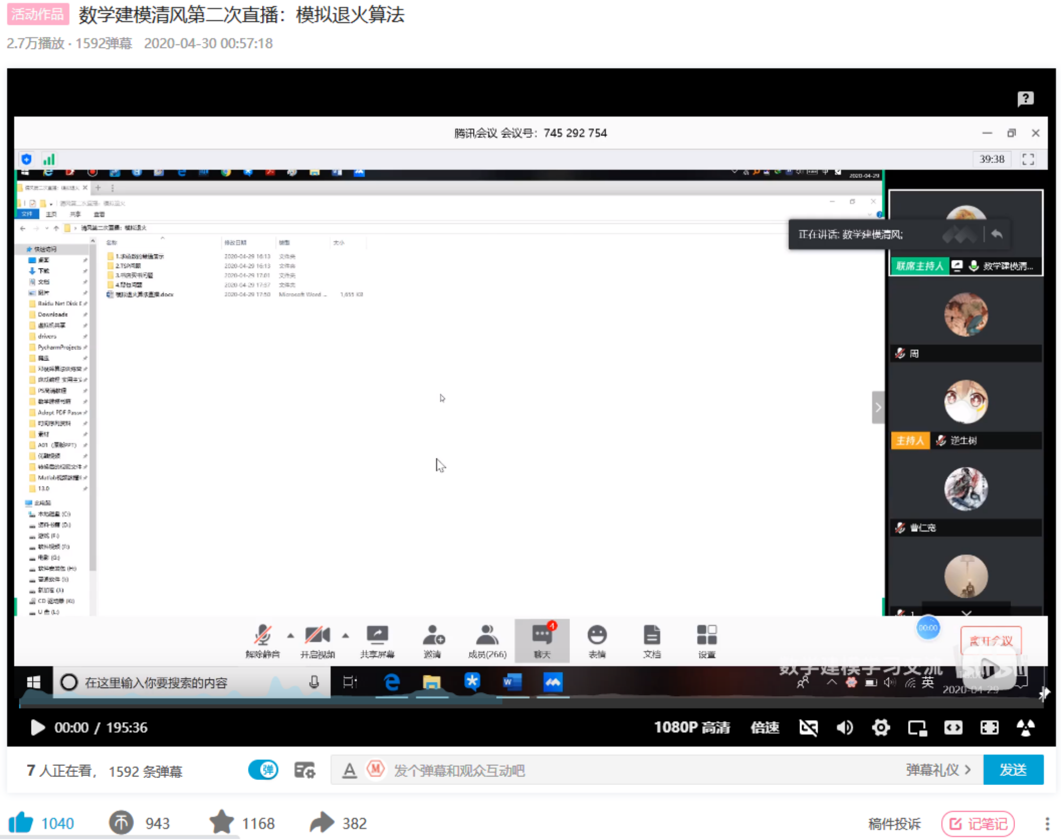
注：本课件配套的视频可在bilibili网站上面免费观看，这是数学建模清风老师讲解的公开课系列。

视频观看地址：https://www.bilibili.com/video/BV1hK41157JL



由于作者水平所限，课件中存在的不妥之处，敬请读者不吝赐教。
如需在博客或者论坛中**引用**本课件中的内容，**请注明来源**，格式如下：

参考资料：数学建模清风第二次直播：模拟退火算法 https://www.bilibili.com/video/BV1hK41157JL

数学建模学习交流

数学建模清风 邀请您参加腾讯会议
会议主题：如何灵活运用模拟退火算法解题？
会议时间：周三 4/29 18:00-19:00

会议 ID：745 292 754
会议容量 300 人，请要参加会议的同学提前进入会议室，人满后会议会被锁定进入不了。
如果到时候有热心的同学帮忙录屏的话可以把录屏发给我，我会放到 B 站上面~
本直播配套的讲义代码等资料明天我会放在公众号《数学建模学习交流》上，大家回复“模拟退火”即可免费获取。

内容：
(1) 介绍模拟退火的核心思想（与蒙特卡罗模拟对比）
(2) 利用上述思想带大家看实际例子（旅行商问题，书店买书问题，背包问题）
(3) 模拟退火需要注意的点

四个问题：（蒙特卡罗模拟解决简单点的情况还行，问题只要一复杂就没办法了）

（1） 求一个给定的函数的最值问题

$$\text{求函数}y = 11\sin x + 7\cos(5x)\text{在}[-3,3]\text{内的最大值}$$

（暴力搜索？如果有 30 个自变量怎么办？）

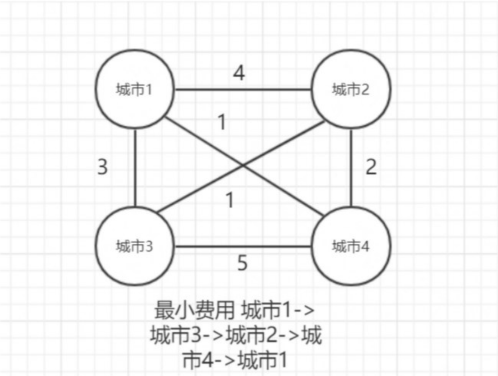
（2） TSP（旅行商问题）

(6) 旅行商问题 (Traveling Salesman Problem, TSP)

一个售货员必须访问n个城市，这n个城市是一个完全图，售货员需要恰好访问所有城市的一次，并且回到最终的城市。

城市于城市之间有一个旅行费用，售货员希望旅行费用之和最少。

完全图：完全图是一个简单的无向图，其中每对不同的顶点之间都恰连有一条边相连。



假设有n个城市，则运算的开支为n!

$n=10 \to 10^6$

$n=15 \to 10^{12}$

$n=20 \to 10^{18}$

$(1亿=10^8)$

有(n-1)!种路径

每一条路径要做n个求和

MATLAB
code7.m

未来学习智能算法时会解决n很大的情形。

（3） 书店买书问题（假设有 m 个书店，n 本书，那么买书的方案数为m^n）

(4) 书店买书问题 (0-1 规划)

某同学要从六家线上商城选购五本书籍 B1, B2, B3, B4, B5，每本书籍在不同商家的售价以及每个商家的单次运费如下表所示，请给该同学制定最省钱的选购方案。

（注：在同一个店买多本书也只会收取一次运费）

	B1	B2	B3	B4	B5	运费
A 商城	18	39	29	48	59	10
B 商城	24	45	23	54	44	15
C 商城	22	45	23	53	53	15
D 商城	28	47	17	57	47	10
E 商城	24	42	24	47	59	10
F 商城	27	48	20	55	53	15

（4） 背包问题（如果有 n 件货物，那么可能性有 2^n 种）

- 有10件货物要从甲地运送到乙地，每件货物的重量(单位：吨)和利润(单位：元)如下表所示。

物品	1	2	3	4	5	6	7	8	9	10
重量(t)	6	3	4	5	1	2	3	5	4	2
利润(元)	540	200	180	350	60	150	280	450	320	120

- 由于只有一辆最大载重为30t的货车能用来运送货物，所以只能选择部分货物进行运送。
- 要求确定运送哪些货物，使得运送这些货物的总利润最大。

上述要解决的问题：

某个目标函数的最值（某一给定的函数、旅行的路程或费用、买书的花费、利润）

（最大值问题通过给目标函数添加负号可以转换为求最小值问题）

求解的通用步骤：

（1） 蒙特卡罗模拟（这里用其求解最值问题）

随机生成很多组解，验证这些解是否满足题目的约束条件，若满足则将其保存到一个"可行集"中，然后计算这个可行集内每个解对应的目标函数值，在里面找到最值即可。

对比穷举法：穷举法是按照某一特定的规则来穷举整个空间，蒙特卡罗模拟是随机搜索，这是两者的区别。共同的缺点：如果解空间的可能情况特别多就 GG

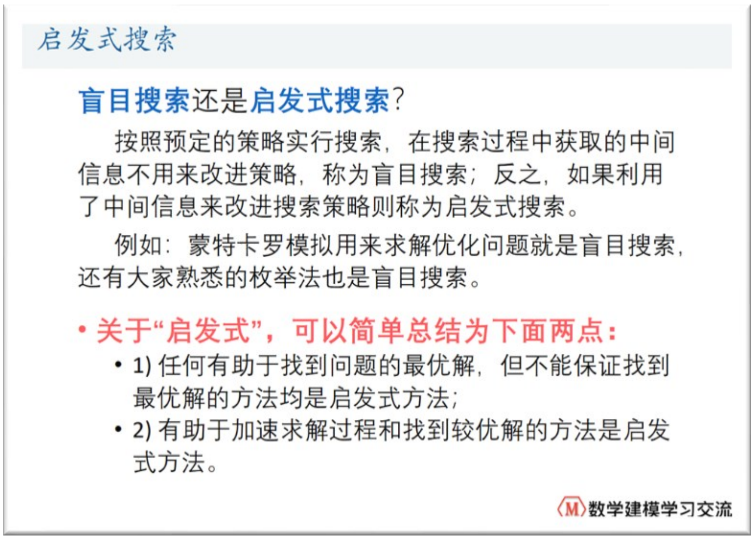
求函数$y = 11\sin x + 7\cos(5x)$在$[-3,3]$内的最大值

蒙特卡罗模拟：随机生成一万个 x: [0.5245 2.2515 -1.0251 1.5441 -2.8954……]

穷举法：x 每次增长一个固定的长度：[-3 -2.9999 -2.9998 -2.9997 ……]

注意：当要求的函数中有 30 个自变量时，上面两种方法都得凉凉

（**NP-Hard**：TSP 有 38 个城市时，对应的计算量是 10^44 次，假设我们的计算机每秒可进行 10 亿次计算，那么穷尽所有解的时间大于 1"亿亿亿"年）

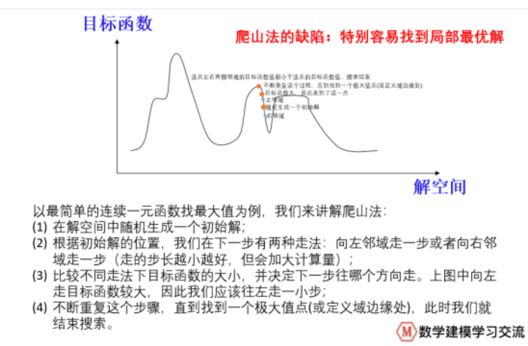


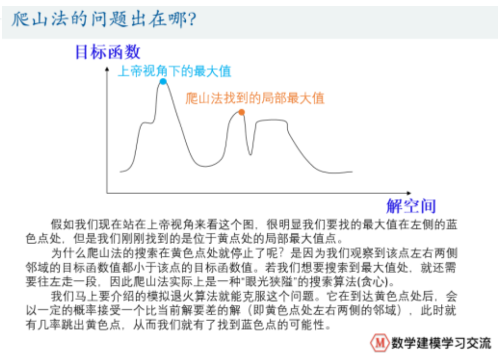
模拟退火算法就是启发式搜索算法中的一种。

（其他你可能感兴趣的启发式搜索算法：粒子群、遗传、蚁群等）

（2） 模拟退火算法

在讲解模拟退火之前，我们先看另一个方法：**爬山法**





启发式算法：在搜索最优解的过程中利用到了原来搜索过程中得到的信息，且这个信息会改进我们的搜索过程。(深刻体会这句话!!!)

思考：爬山法是启发式算法吗？

是的哦~

爬山法的缺陷：特别容易找到局部最优解；

那么模拟退火算法怎么克服这个缺点呢？

举例来说明：

求函数$y = 11\sin x + 7\cos(5x)$在$[-3,3]$内的最大值

旧的解：x_i,对应的函数值为$f(x_i)$

新的解：在x_i“附近”随机找一个新的解x_j，对应的函数值为$f(x_j)$

(1) 如果$f(x_j) > f(x_i)$，则接受新解x_j;

(2) 如果$f(x_j) \le f(x_i)$，那么我们应该直接拒绝x_j吗？ （注：等号跟着哪一边无所谓）

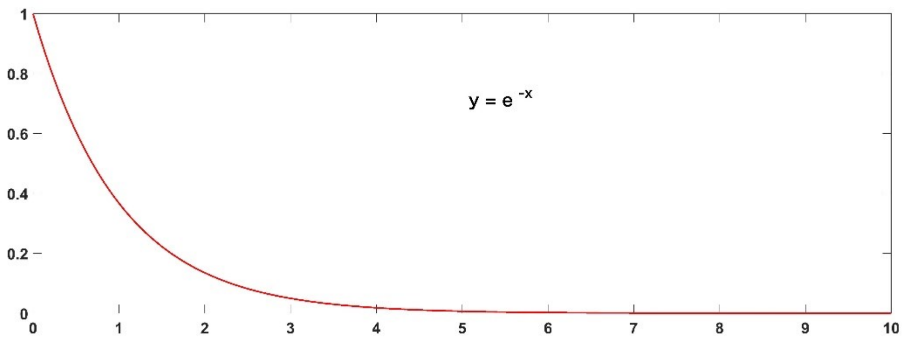
答案是不应该，如果拒绝了那不就是登山法了吗?

解决方法：为了不直接拒绝x_j，我们定义一个接受的概率 p，p 位于[0,1]之间，且$f(x_j)$和$f(x_i)$越接近，p 越大。**(这种想法很直观，新解和旧解的函数值越接近，我们就越愿意接受新解)**

即 p 正比于$\frac{1}{|f(x_j)-f(x_i)|}$，记为$p \propto \frac{1}{|f(x_j)-f(x_i)|}$ （在上面的问题中：$|f(x_j)-f(x_i)| = f(x_i)-f(x_j)$）

思考：概率 p 位于[0,1]之间，哪个常见的函数值域是位于[0,1]之间的?

不唯一，但有一个函数形式很简单：$y = e^{-x}(x \ge 0)$，关于 x 单调递减



所以我们可以假设：$p \propto e^{-|f(x_j)-f(x_i)|}$

（选择题）接受新解的概率 p 越大意味着在解空间中搜索的范围______?

A: 越大　　B：越小

原因：接受新解的概率 p 越大意味着我们的搜索更容易接受不好的解，这时候就相当于增加了我们搜索的范围。

（选择题）假设我们将搜索过程看作一个“工序”，那么搜索前期我们搜索的范围应该尽量的_____，搜索后期我们搜索的范围应该尽量的_____。

A: 大、小　　B：小、大

原因：在搜索前期，我们要尽可能的扩大我们的搜索范围，这样能够避免陷入局部最优解；在搜索后期我们要倾向于局部搜索。(“渣男”套路：前期广撒网，后期精准打击)。

我们对式子$\mathbf{p} \propto e^{-|f(x_j)-f(x_i)|}$进行一个变形：

$$\mathbf{p}_t \propto e^{-|f(x_j)-f(x_i)| \times C_t}$$

（选择题）其中： C_t 可以看作一个和时间有关的系数，因此在搜索过程中，我们接受新解的概率 p 就和时间有关啦，显然：时间越长，C_t 越_____。

A：大　　B：小

原因：搜索前期 t 较小，我们希望搜索的范围大，即更倾向于接受新解，那么对应的 p 就应该大一点，而C_t与 p 负相关，因此C_t应该小一点；　搜索后期我们希望 p 较小，那么C_t应该大一点，因此我们可以下结论：C_t关于 t 递增　。

我们的搜索过程(假设求**最大值**问题)可以用下面这个简单的流程表示：

（1）　随机生成一个解 A，计算解 A 对应的目标函数值 f(A)

（2）　在 A 附近随机生成一个解 B，计算解 B 对应的目标函数值 f(B)

（3）　如果 f(B)>f(A)，则将解 B 赋值给解 A，然后重复上面步骤（爬山法的思想）；
　　　如果 f(B)≤f(A)，那么我们计算接受 B 的概率$p_t = e^{-|f(B)-f(A)| \times C_t}$，然后我们生成一个[0,1]之间的随机数 r，如果 r<p，我们就将解 B 赋值给解 A，然后重复上面的步骤；否则我们返回第(2)步，在原来的 A 附近再重新生成一个解 B，然后继续下去。

上面这个流程里面有几个我们需要去研究的地方：

（1）　如果这个优化问题有约束条件怎么办呀？

（2）　这个C_t怎么设置呀？

（3）　这个搜索过程什么时候结束呀？

（4）　怎么样在 A 附近随机生成一个新解 B 呀？

（后三个问题我们等会再来依次看）

第一个问题：连麦回答环节

C_t的设置：模拟退火(目前已知：$p_t = e^{-|f(B)-f(A)| \times C_t}$，且$C_t$关于t递增)

退火是一种金属热处理工艺，指的是将金属缓慢加热到一定温度，保持足够时间，然后以适宜速度冷却。目的是降低硬度，改善切削加工性；降低残余应力，稳定尺寸，减少变形与裂纹倾向；细化晶粒，调整组织，消除组织缺陷。准确的说，退火是一种对材料的热处理工艺，包括金属材料、非金属材料。而且新材料的退火目的也与传统金属退火存在异同。

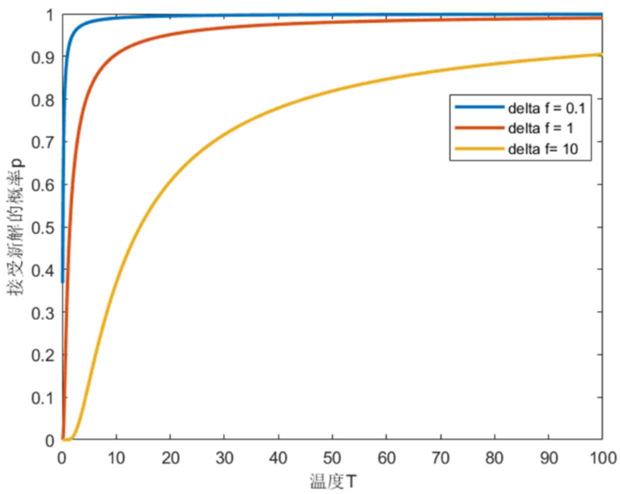
定义初始温度$T_0 = 100$，温度下降的公式为：$T_{t+1} = \alpha T_t$，α常取 0.95，那么时刻 t 时的温度：$T_t = \alpha^t T_0 = 100 \times 0.95^t$.

取$C_t = \frac{1}{T_t} = \frac{1}{100 \times 0.95^t}$，那么$p_t = e^{-\frac{|f(B)-f(A)|}{T_t}} = e^{-\frac{|f(B)-f(A)|}{100 \times 0.95^t}}$

注意：这里取倒数是为了保证C_t关于t递增。

$$p_t = e^{-\frac{|f(B)-f(A)|}{T_t}}$$

记$\Delta f = |f(B)-f(A)|$, 那么 p 到底有多大呢?



（上图从右往左看就是退火过程）

（1）温度一定时， Δf越小，概率越大，即目标函数相差越小接受的可能性越大。
（2）Δf一定时，温度越高，概率越大，即搜索前期温度较高时更有可能接受新解。

回到搜索的流程：

我们的搜索过程(假设求最大值问题)可以用下面这个简单的流程表示：

（1） 随机生成一个解 A，计算解 A 对应的目标函数值 f(A)

（2） 在 A 附近随机生成一个解 B，计算解 B 对应的目标函数值 f(B)

（3） 如果 f(B)>f(A)，则将解 B 赋值给解 A，然后重复上面步骤（爬山法的思想）；

如果 f(B)≤f(A)，那么我们计算接受 B 的概率$p_t = e^{-\frac{|f(B)-f(A)|}{100\times 0.95^t}}$,然后我们生成一个[0,1]之间的随机数 r，如果 r<p，我们就将解 B 赋值给解 A，然后重复上面的步骤；否则我们返回第(2)步，在原来的 A 附近再重新生成一个解 B，然后继续下去。

这里的 t 在编程中怎么实现？很简单，可以看成我们迭代的次数。（循环）

为了保证搜索过程的彻底，在同一温度下（同一个小 t）我们需要进行多次搜索（例如重复上面的流程 500 次）；之后我们降低温度，然后再来进行新的一轮搜索。

讨论：如果我们编程的话，上述流程中有几层循环？

什么时候停止搜索？

很多种准则：（1）达到指定的迭代次数，例如迭代 200 次；（2）达到指定的温度，例如温度小于 0.000001；（3）我们找到的最优解连续 M（例如 30 次）次迭代都没有变化

这里演示 live_code1.m

最后剩下一个问题：

怎么样在 A 附近随机生成一个新解 B 呀？即新解应该怎么样生成？

答案：没有统一规定，具体问题具体分析，我们来看例题体会吧。

题目 1：求给定函数的最大值或者最小值

我们以n元函数求最值为例，来介绍新解的产生规则：

假设当前解为：$(x_1, x_2, \cdots, x_n)$

我们首先生成一组随机数$(y_1, y_2, \cdots, y_n)$，其中y_i服从$N(0,1)$

接下来我们再计算$(z_1, z_2, \cdots, z_n)$，其中$z_i = y_i / \sqrt{y_1^2 + y_2^2 + \cdots + y_n^2}$，

对于每一个$i \in [1, n]$，进行下面的操作：

(1) 计算：$x_i^{new} = x_i + T \times z_i$，$T$是当前的温度（也可以使用$x_i^{new} = x_i + \sqrt{T} \times z_i$这个公式）

(2) 接下来检查x_i^{new}是否位于上下界内，即是否满足$lb_i \le x_i^{new} \le ub_i$这个约束

(a) 如果$lb_i \le x_i^{new} \le ub_i$满足，则直接令$x_j = x_i^{new}$即可

(b) 如果$x_i^{new} < lb_i$，则$x_j = r \times lb_i + (1-r) \times x_i$，这里$r$是区间$[0,1]$上均匀分布的随机数

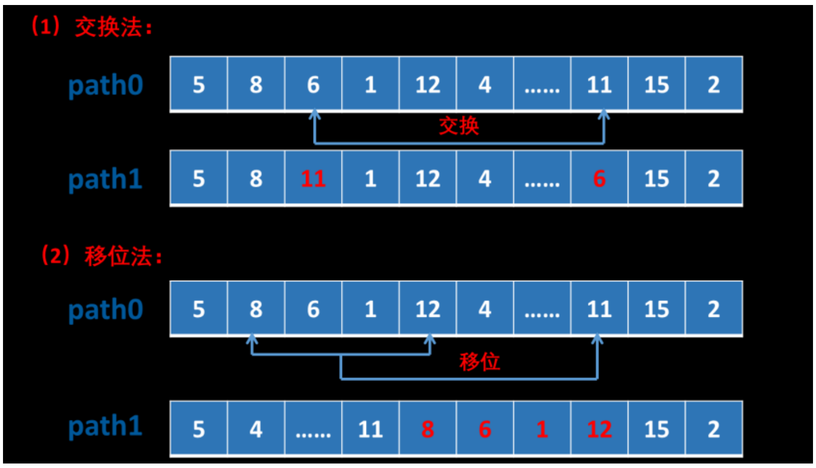
(c) 如果$x_i^{new} > ub_i$，则$x_j = r \times ub_i + (1-r) \times x_i$，这里$r$是区间$[0,1]$上均匀分布的随机数

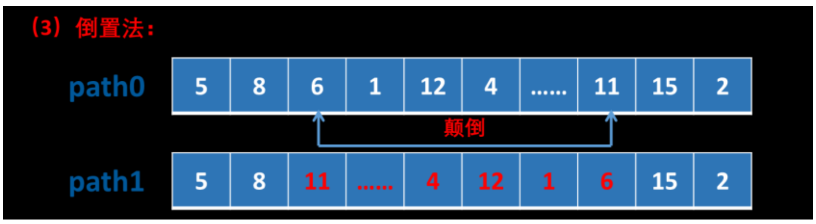
注意：新解的产生规则在学术上没有统一的规定，我这里参考的是 Matlab 内置函数的那种定义方法。

题目 2：旅行商问题

旅行商问题（**Travelling salesman problem, TSP**）是这样一个问题：给定一系列城市和每对城市之间的距离，求解访问每一座城市一次并回到起始城市的最短回路。

三种产生新解的方法 **参考：苗卉，杨韬．旅行商问题(TSP)的改进模拟退火算法[J]．微计算机信息, 2007**



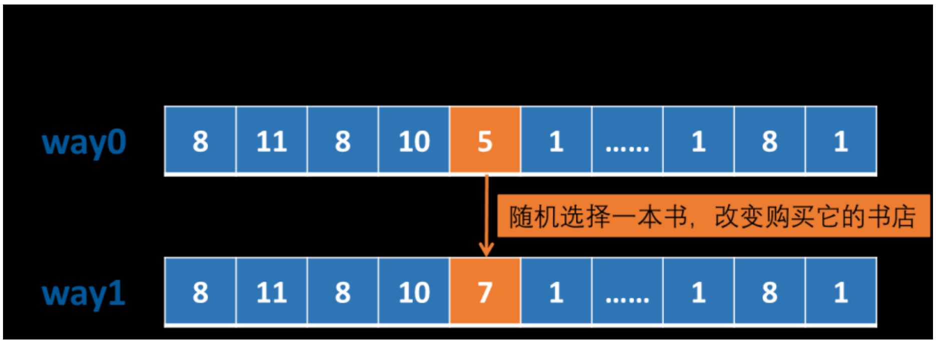


题目 3：书店买书问题

15 家书店购买 20 本书，每个书店的运费在最后一列，怎么选择买书的方案？

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19	B20	运费
S1	31	31	41	21	25	28	23	34	38	29	38	33	32	24	23	20	23	26	21	32	10
S2	40	27	38	26	23	29	24	22	37	29	32	34	31	27	31	22	26	27	25	27	10
S3	35	25	41	20	26	21	37	24	34	22	42	31	37	26	28	23	23	21	26	28	14
S4	33	26	22	29	38	25	34	32	34	24	27	25	26	31	39	34	21	21	41	34	7
S5	33	29	36	24	21	24	33	28	25	29	24	26	26	29	37	24	25	25	32	27	12
S6	25	32	20	21	20	32	42	22	33	24	35	28	38	26	34	21	39	25	40	23	5
S7	35	22	35	29	29	26	38	30	27	21	25	30	33	32	30	32	25	23	26	23	10
S8	36	22	39	26	34	25	32	23	35	29	20	32	34	31	25	24	38	25	29	25	8
S9	32	23	22	21	27	22	20	30	27	24	41	27	33	27	29	22	31	26	25	24	14
S10	27	28	36	22	38	27	29	33	29	25	29	33	34	25	24	22	37	27	42	30	9
S11	39	28	26	27	37	28	23	31	35	27	30	28	20	32	31	21	32	31	43	21	12
S12	22	28	38	33	40	23	43	30	35	24	23	26	36	23	34	24	40	24	41	30	6
S13	30	25	27	32	27	30	40	27	36	22	30	29	21	32	41	33	33	29	31	31	11
S14	34	21	27	29	25	21	36	33	21	28	21	30	35	22	22	24	40	27	25	23	5
S15	31	27	24	25	39	23	40	30	22	28	38	31	21	29	21	25	40	22	31	35	9

产生新解的方法：



因为没有文献参考，我只想了这一种产生新解的方法，要是有同学使用其他产生新解的方法得到了更小的花费(<466)，欢迎和我交流，(づ￣ 3￣)づ。

题目 4：背包问题

- 有10件货物要从甲地运送到乙地，每件货物的重量(单位：吨)和利润(单位：元)如下表所示。

物品	1	2	3	4	5	6	7	8	9	10
重量(t)	6	3	4	5	1	2	3	5	4	2
利润(元)	540	200	180	350	60	150	280	450	320	120

- 由于只有一辆最大载重为30t的货车能用来运送货物，所以只能选择部分货物进行运送。
- 要求确定运送哪些货物，使得运送这些货物的总利润最大。

参考论文：梁国宏, 张生, 黄辉, et al. 一种改进的模拟退火算法求解 0-1 背包问题[J]. 广西民族大学学报(自然科学版), 2007(03):97-99.

课后习题：

（1） 自己写一个模拟退火的代码，求解下面这个函数的最大值

$$y = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2),\ x_1 \in [-3, 12.1],\ x_2 \in [4.1, 5.8]$$

参考的最大值：38.8503　　提示：这个函数有非常多的局部最大值，怎么设置合适的退火参数需要不断尝试。

（2） 在书店买书问题中：加入下面这个规则：“如果在同一个书店的消费额不低于 88 元，那么这个书店包邮，即在这个书店买书不用出运费了”，请你修改代码对该问题重新求解。

这个问题并不难，只需要修改计算花费的函数“calculate_money”，当然你也可以改进产生新解的方式。我求的最优解为 439(●ˇ∀ˇ●)

（3） 在经典的 TSP 问题中，假设有 n 个城市，我们要从某一个城市出发，依次访问完所有城市后再回到起点处。现在，我们探究一个类似的问题：假设起点城市 M 和终点城市 N 是给定的，要我们求出一个从 M 出发依次经过所有城市并最终到达 N 的最短路径，即不再要求我们回到起点 M 处。请大家用模拟退火来求解这个问题。

提示：可以把起点城市放在下标为 1 的位置，将终点城市放在下标为 n 的位置。当起点为 5 且终点为 20 时，我得到的最短路径为 6636。

本讲义内容以及配套的代码可能有不完善或者错误，文责自负

模拟退火算法有哪些要注意的点：

(1) 以上我们的讨论都没有涉及到一个核心的问题？模拟退火算法一定能找到最优解吗？可以利用随机过程中的马尔科夫过程证明，在一定条件下，当温度降为 0 时，最后的状态一定是全局最优解。这样就能够为模拟退火算法在求解最优化问题提供理论支持。

参考：康立山等 1994 年编写的《非数值并行算法（第一册）模拟退火算法》

(2) 对参数的选择（例如初始的温度）以及退火流程的设置需要经验，参数设置的不恰当得到的解可能会很差。因此**我们可以多次尝试不同的参数，观察求解时间和求解结果，以此来对参数进行修改。**

参考：卓金武等 《MATLAB 在数学建模中的应用》，北京航空航天大学出版社, 2011, P85

(3) 新解的构造至关重要，新解不能距离旧解“太远”，否则旧解的信息不能被新解反映；同时新解也不能距离旧解“太近”，否则容易陷入局部最优。在实际应用中，我们可以先去知网搜搜有没有类似的问题，看看别人是如何构造新解的，如果实在是没有相关的研究，我们再去构造自己的新解。

(4) 如果优化问题中的约束非常多，那么这时候构造新解就有很大的**技巧性**。如果构造的新解不在约束条件内，虽然我们可以重新构造新解，但这样会增加我们的计算量。

(5) 模拟退火和其他智能算法混合使用，比如粒子群、遗传等