



数学实验

实验十三：现代优化算法

翟晓雅

Email: xiaoyazhai@ustc.edu.cn

Homepage: <https://xiaoyazhai.github.io/>

实验目的

- 了解现代优化算法
- 学会现代序列算法的MATLAB 编程
- 针对实际问题学会利用现代优化算法进行建模并求解

13.1 算法背景

对于一个求函数最大值的优化问题（求最小值也类同），一般可描述为下述数学规划模型：

$$\begin{array}{ll} \max & f(X) & (1-1) \\ \text{s.t.} & X \in R & (1-2) \\ & R \subseteq U & (1-3) \end{array}$$

其中：

$X = [x_1, x_2, \dots, x_n]^T$ 为决策变量， $f(X)$ 为目标函数，
式(1-2)、(1-3)为约束条件， U 是基本空间， R 是 U 的一个子集。

满足约束条件的解 X 称为可行解；

集合 R 表示由所有满足约束条件的解所组成的一个集合，叫做可行解集合。

13.1 算法背景

- 上述最优化问题，目标函数和约束条件种类繁多，有的是线性的，有的是非线性的；有的是连续的，有的是离散的；有的是单峰值的，有的是多峰值的。
- 求最优解或近似最优解的方法主要有三种：
枚举法、启发式算法和搜索算法。
- 随着问题种类的不同，以及问题规模的扩大，要寻求到一种能以有限的代价来解决上述最优化问题的通用方法仍是个难题。

13.1 算法背景

局部优化算法&全局性优化算法

如果决策问题是一个**凸问题**，可以利用**线性规划**、**非线性规划**等求解。然而大量的实际问题是**非凸问题**，需要在大量的局部极优解中寻找全局最优解。

习惯上，将优化算法分为两类：**局部优化算法**和**全局性优化算法**。前者可以称为**经典优化算法**，后者习惯上称为**现代优化算法**。

13.1 算法背景

局部优化算法&全局性优化算法

局部优化算法主要用于解决凸问题或单峰问题，通常使用确定性搜索策略，比如单纯形法、梯度下降法、爬山法、贪心法等，其基本思想是在状态转移过程中，只接受更好的状态，拒绝恶化的状态。

13.1 算法背景

局部优化算法&全局性优化算法

全局性优化算法主要用于求解**非凸问题**或**多峰问题**，通常使用**概率性搜索策略**，即**状态转移规则**。从任一解出发，利用适当的状态转移规则和合理的概率性状态接收规则搜索新的更优点，在确定的时间或搜索次数之内停止算法。

13.1 算法背景

现代优化算法80 年代初兴起的启发式算法。这些算法包括：

- 禁忌搜索 (tabu search)
- 模拟退火 (simulated annealing)
- 遗传算法 (genetic algorithms)



NP-hard组合优化问题

13.1 算法背景

引入旅行商问题 (Travelling salesman problem, TSP)

- 最早可追溯为1759年Euler提出的骑士旅行问题。
- 该问题是否存在一个有效的通用的求解方法仍然是一个开放性问题。

13.1 算法背景

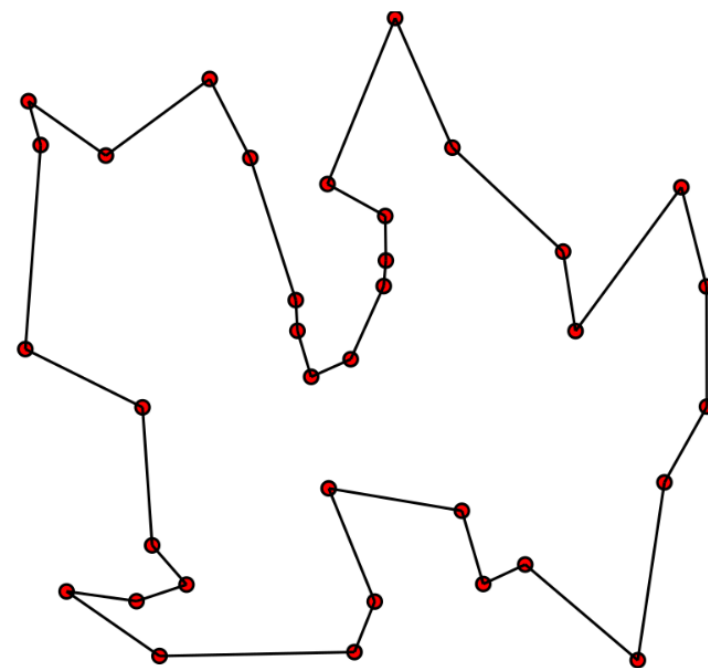
旅行商问题 (Travelling salesman problem, TSP)

- 问题：给定一系列城市和每对城市之间的距离，求解访问每座城市一次并回到起始城市的最短回路。
- 组合优化中的一个NP困难问题;
- TSP算法最坏情况下的时间复杂度随着城市数量的增多而成超多项式（可能是指数）级别增长。

13.1 算法背景

旅行商问题 (Travelling salesman problem, TSP)

- 可以用无向加权图来对TSP建模。
- 城市是图的顶点，道路是图的边，道路的距离就是该边的长度。它是起点和终点都在一个特定顶点，访问每个顶点恰好一次的最小化问题。
- 通常，该模型是一个完全图（即每对顶点由一条边连接）。如果两个城市之间不存在路径，则增加一条非常长的边就可以变为完全图，而不影响计算最优回路。

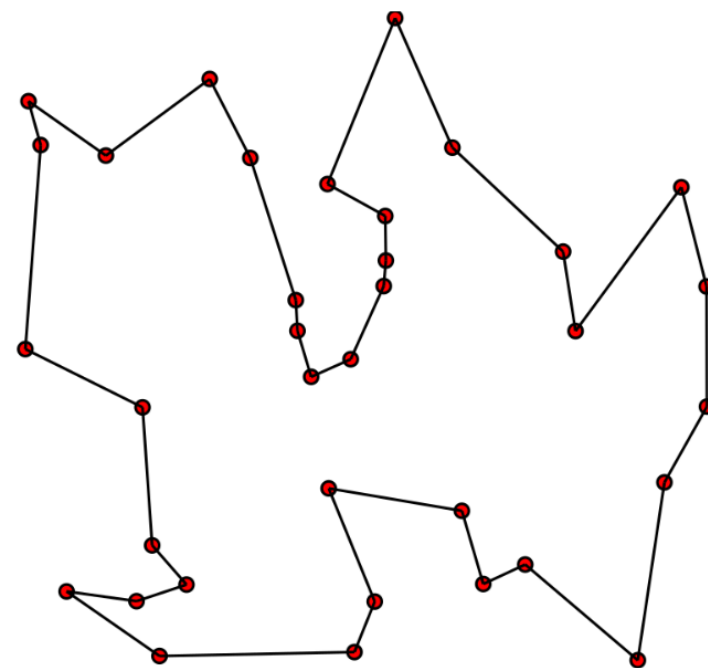


13.1 算法背景

旅行商问题 (Travelling salesman problem, TSP)

在**对称TSP**问题中，两座城市之间来回的距离是相等的，形成一个无向图。这种对称性将解的数量减少了一半。

在**非对称TSP**问题中，可能不是双向的路径都存在，或是来回的距离不同，形成了有向图。交通事故、单行道和出发与到达某些城市机票价格不同等都是打破这种对称性的例子。



13.2 禁忌搜索算法 (TABU search)

禁忌，就是禁止重复前面的工作。

-为了回避局部邻域搜索陷入局部最优的主要不足，采用一个**禁忌表**记录已经达到过的局部最优点，在下一次搜索中，利用禁忌表中的信息不再或者有选择地搜索这些点，以此来跳出局部最优点。

Glover在1986年首次提出这一概念，进而形成一套完整算法。

13.2 禁忌搜索算法 (TABU search)

Tabu算法由几个基本要素的组合：

- **邻域**与一般优化技术中的定义一致；
- **Tabu表**是一个或数个数据序列，是对先前的数步搜索所作的记录，记录的方式有很多，记录的长度也是可变的，选取的好坏直接影响算法的效率；
- **评价函数**通常就是问题的目标函数或它的某种变换形式，用于对一个移动作出评价。
由Tabu表和评价函数可以构造一种Tabu条件，若新点满足Tabu条件则接受，否则拒绝，直至迭代终止。

13.2 禁忌搜索算法 (TABU search)

邻域的概念

- (1) 函数优化问题: 邻域($N(x)$)通常定义为在给定距离空间内, 以一点(x)为中心的一个球体。
- (2) 组合优化问题:

$$N : x \in X \rightarrow N(x) \in 2^X$$

且 $x \in N(x)$, 称为一个邻域映射, 其中 2^X 表示 X 所有子集组成的集合。

$N(x)$ 称为 x 的邻域, $y \in N(x)$ 称为 x 的一个邻居。

13.2 禁忌搜索算法 (TABU search)

➤ 邻域及邻域移动

- ① 定义邻域移动 s ，例如，在函数优化问题中邻域移动可以定义为给定步长和移动方向；在组合优化问题中邻域移动可以定义为某种序列置换。
- ② 邻域是由当前解 x 及其通过定义的邻域移动能够达到的所有解构成的集合。

13.2 禁忌搜索算法 (TABU search)

➤ 禁忌表

禁忌表(T表)的作用：防止搜索出现循环

- ① 将移动、移动分量或适值作为禁忌对象
- ② 表的长度称为Tabu-Size，可以用来控制局域搜索和广域搜索
- ③ 表是动态更新的——把最新的解记入，最老的解从表中释放(解禁)

13.2 禁忌搜索算法 (TABU search)

➤ 选择策略

选择策略的作用：保证TS具有跳出局优的能力，当前解 x 每一步总是移动到邻域 $N(x)$ 中未被禁忌的最优解，即若

$$C(s_k(x)) = Opt \{ C(s(x)), s(x) \in N(x) \setminus T \}$$

则令 $x = S_k(x)$ ，本次移动到邻域 $N(x)$ 中未被禁忌的最优解 $S_k(x)$ 。

13.2 禁忌搜索算法 (TABU search)

例：TSP问题解的一种表示方法为 $D = \{x = (i_1, i_2, \dots, i_n) | i_1, i_2, \dots, i_n \text{ 是 } 1, 2, \dots, n \text{ 的排列}\}$ ，定义它的邻域映射为 2-opt，即 x 中的两个元素进行对换， $N(x)$ 中共包含 x 的 $C_n^2 = n(n-1)/2$ 个邻居和 x 本身。

例如： $x = (1, 2, 3, 4)$ ，则 $C_4^2 = 6$ ，

$$N(x) = \{(1, 2, 3, 4), (2, 1, 3, 4), (3, 2, 1, 4), (4, 2, 3, 1), (1, 3, 2, 4), (1, 4, 3, 2), (1, 2, 4, 3)\}$$

13.2 禁忌搜索算法 (TABU search)

定义邻域移动为：位值加1或减1

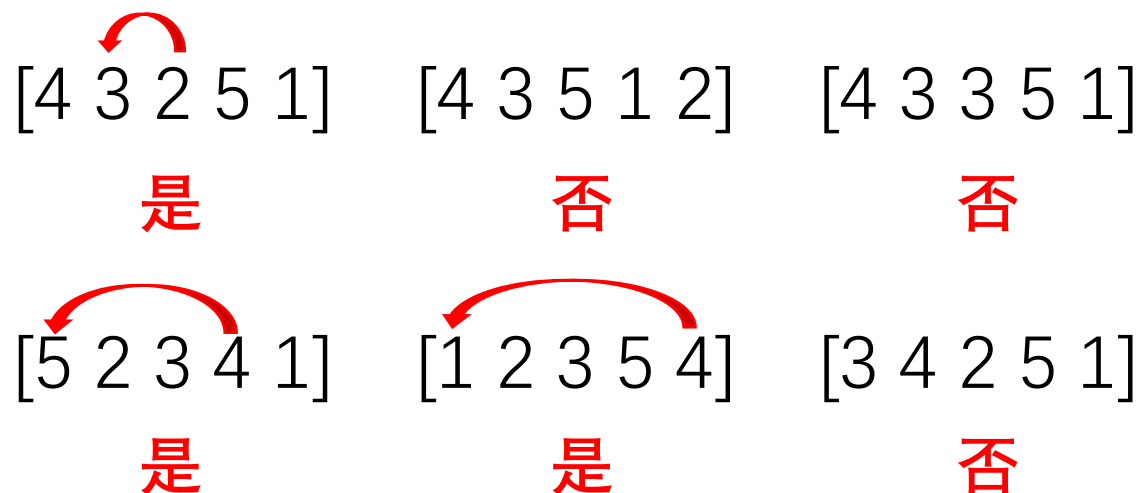
对整数编码[2 2 3 5 3]，下列编码是否在其邻域内：

[2 <u>3</u> 3 5 3]	[2 <u>3</u> <u>2</u> 5 3]	[2 2 3 5 <u>5</u>]
是	否	否
[2 2 3 <u>4</u> 3]	[2 2 <u>2</u> 5 3]	[2 2 3 <u>4</u> <u>4</u>]
是	是	否

13.2 禁忌搜索算法 (TABU search)

定义邻域移动为：2-Opt

对顺序编码[4 2 3 5 1]， 下列编码是否在其邻域内：



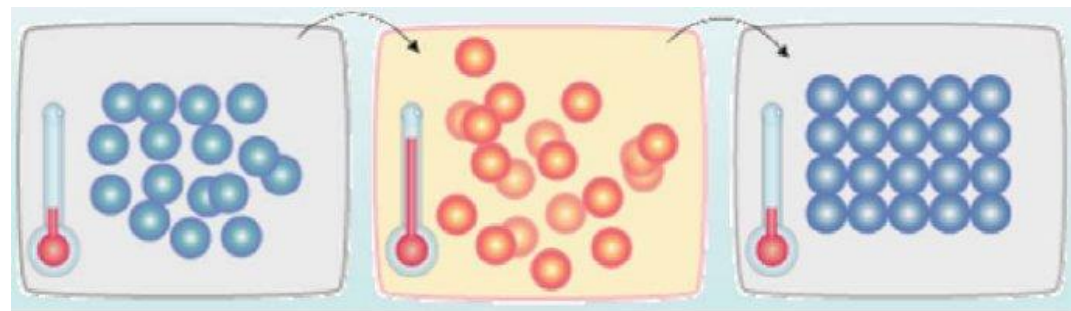
13.3 模拟退火算法

- 退火（Annealing），在冶金学或材料工程，是一种改变材料微结构且进而改变如硬度和强度等机械性质的热处理。**过程为将金属加温到高于再结晶温度的某一温度并维持此温度一段时间，再将其缓慢冷却。**退火的功用在于恢复该金属因冷加工而降低的性质，增加柔软性、延展性和韧性，并释放内部残留应力、以及产生特定的显微结构。

13.3 模拟退火算法

统计力学表明材料中粒子的不同结构对应于粒子的不同能量水平。

- 在高温条件下，粒子的能量较高，可以自由运动和重新排列。
- 在低温条件下，粒子能量较低。如果从高温开始，非常缓慢地降温（这个过程被称为退火），粒子就可以在每个温度下达到热平衡。
- 当系统完全被冷却时，最终形成处于低能状态的晶体。



加温过程

等温过程

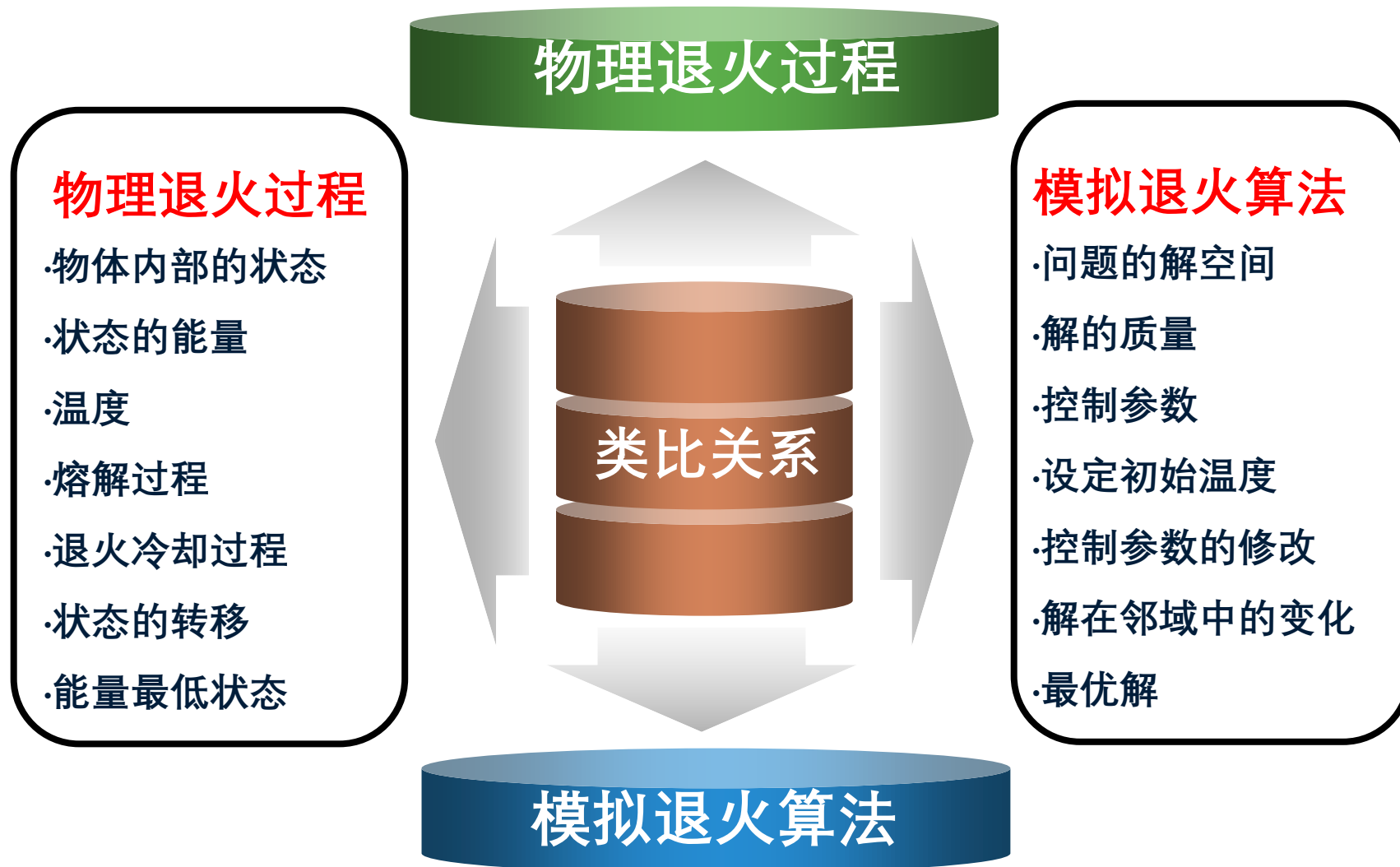
冷却过程

13.3 模拟退火算法

基本思想:

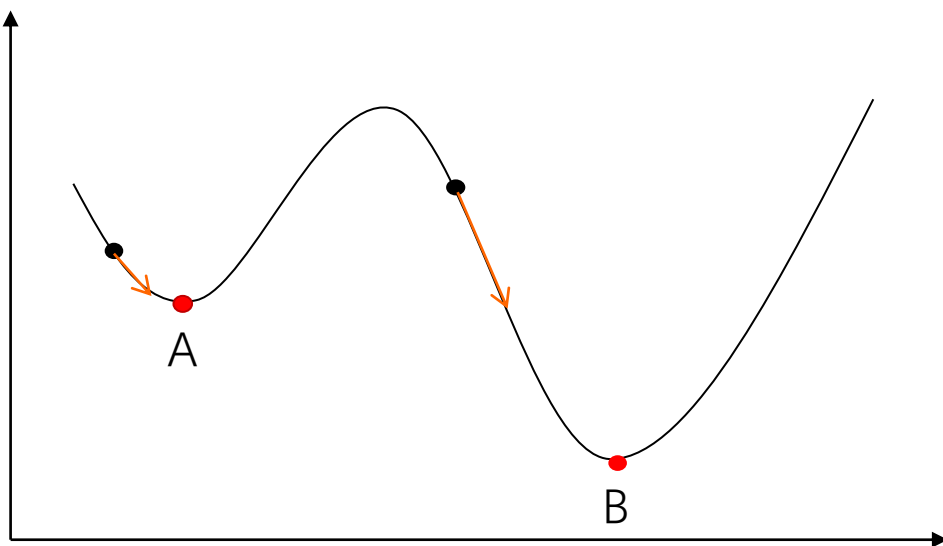
- 模拟退火算法(Simulated Annealing Algorithm, SAA)是一种基于迭代求解策略的随机寻优算法, 其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。
- 模拟退火算法从某一较高初温出发, 伴随温度参数的不断下降, 结合概率突跳特性在解空间中随机寻找目标函数的全局最优解, 即当系统找到局部最优解时能概率性地跳出并最终趋于全局最优。

13.3 模拟退火算法



13.3 模拟退火算法

模拟退火算法**以一定的概率**来接受一个比当前解要差的解，因此有可能会跳出这个局部的最优解，达到全局的最优解。



13.3 模拟退火算法

Metropolis准则

模拟退火算法采用*Metropolis*接受准则——以概率接受新状态

若在温度 T ，当前状态 $i \rightarrow$ 新状态 j

若 $E_j < E_i$ ，则接受 j 为当前状态；

否则，若概率 $p = \exp[-(E_j - E_i)/kT]$ 大于 $[0,1)$ 区间的随机数，则仍接受状态 j 为当前状态；若不成立则保留状态 i 为当前状态。

13.3 模拟退火算法

SAA机理:

- 优化问题的解视为固体的状态;
- 随机给定优化问题的初始解;
- 给定初始温度;
- 根据当前的解产生新的解;
- 依据Metropolis准则对两个解进行取舍;
- 重复以上两步直到达到热平衡;
- 降低温度继续上述过程直到温度降到最低, 最后的状态就认为是问题的解。

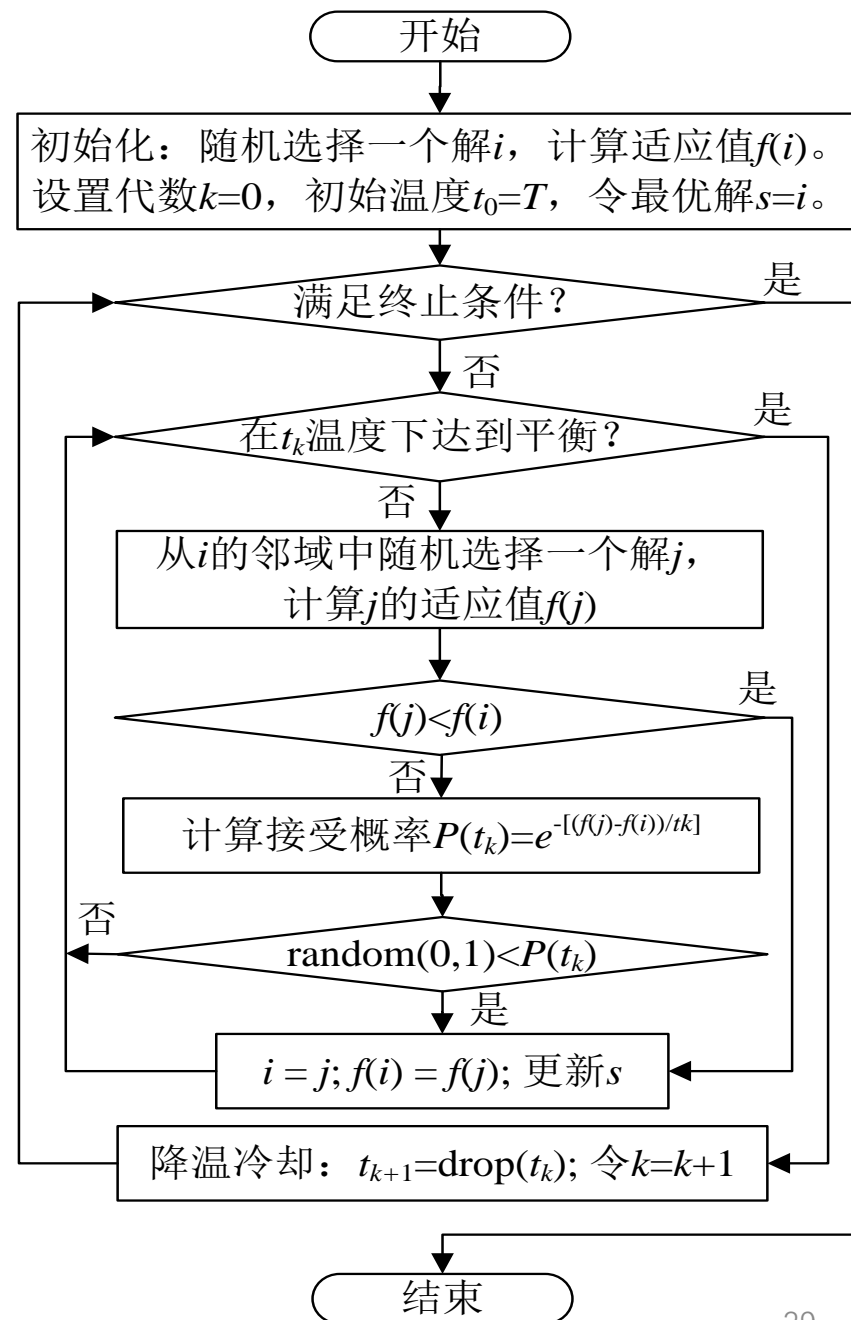
总的来说就是:

从选定的初始解开始, 重复进行“产生新解----计算目标函数差-----判断是否接受新解-----接受(或舍弃)新解”这四个任务的实验, 不断对当前解迭代, 从而达到使目标函数最优的执行过程。

13.3 模拟退火算法

实现步骤

- 1) 随机产生一个初始解 i ，令最优解 $S=i$ ，并计算目标函数值 $J(i)$;
- 2) 设置初始温度 $T(0)=T_0$;
- 3) 判断 $T > T_{min}$ ，满足则对当前最优解 S ，按照某一邻域函数，产生一新的解 j 。计算新的目标函数值 $J(j)$ ，并计算目标函数值的增量 $\Delta E = J(j) - J(i)$ 。
- 4) 如果 $\Delta E < 0$ ，则接受新解，令 $S = j$ ；
如果 $\Delta E > 0$ ，则 $p = \exp(-\Delta E / TK)$ ；
并判断 $p > \text{random}[0,1]$ ，成立则接受新解 $S = j$ ；否则 $S=i$ 。
- 5) 降温 $T=T*r, 0<r<1$ ，回到步骤3)
- 6) 若 $T \leq T_{min}$ ，停止搜索，得到最优解 S 。



13.3 模拟退火算法

1) 优点:

- 可以保证全局最优, 特别适合组合优化问题
- 可以随机选择初始解
- 对问题本身没有特别要求, 不会因为问题实例的改变影响性能
- 简单易行, 通用性好

2) 缺点:

模拟退火算法在求解规模较大的实际问题时, 往往存在以下缺点:

- (1) 收敛速度比较慢。
- (2) 尽管理论上只要计算时间足够长, 模拟退火法就可以保证以概率1收敛于全局最优点。但是在实际算法的实现过程中, 由于计算速度和时间的限制, 在优化效果和计算时间二者之间存在矛盾, 因而难以保证计算结果为全局最优点, 优化效果不甚理想。
- (3) 在每一温度下很难判定是否达到了平衡状态

13.3 模拟退火算法

敌方 100 个目标的经度、纬度如表 1 所示。

我方有一个基地，经度和纬度为(70,40). 假设我方飞机的速度为 1000 公里/小时。我方派一架飞机从基地出发，侦察完敌方所有目标，再返回原来的基地。在敌方每一目标点的侦察时间不计，求该架飞机所花费的时间（假设我方飞机巡航时间可以充分长）。

表 1 经度和纬度数据表

经度	纬度	经度	纬度	经度	纬度	经度	纬度
53.7121	15.3046	51.1758	0.0322	46.3253	28.2753	30.3313	6.9348
56.5432	21.4188	10.8198	16.2529	22.7891	23.1045	10.1584	12.4819
20.1050	15.4562	1.9451	0.2057	26.4951	22.1221	31.4847	8.9640
26.2418	18.1760	44.0356	13.5401	28.9836	25.9879	38.4722	20.1731
28.2694	29.0011	32.1910	5.8699	36.4863	29.7284	0.9718	28.1477
8.9586	24.6635	16.5618	23.6143	10.5597	15.1178	50.2111	10.2944
8.1519	9.5325	22.1075	18.5569	0.1215	18.8726	48.2077	16.8889
31.9499	17.6309	0.7732	0.4656	47.4134	23.7783	41.8671	3.5667
43.5474	3.9061	53.3524	26.7256	30.8165	13.4595	27.7133	5.0706
23.9222	7.6306	51.9612	22.8511	12.7938	15.7307	4.9568	8.3669
21.5051	24.0909	15.2548	27.2111	6.2070	5.1442	49.2430	16.7044
17.1168	20.0354	34.1688	22.7571	9.4402	3.9200	11.5812	14.5677
52.1181	0.4088	9.5559	11.4219	24.4509	6.5634	26.7213	28.5667
37.5848	16.8474	35.6619	9.9333	24.4654	3.1644	0.7775	6.9576
14.4703	13.6368	19.8660	15.1224	3.1616	4.2428	18.5245	14.3598
58.6849	27.1485	39.5168	16.9371	56.5089	13.7090	52.5211	15.7957
38.4300	8.4648	51.8181	23.0159	8.9983	23.6440	50.1156	23.7816
13.7909	1.9510	34.0574	23.3960	23.0624	8.4319	19.9857	5.7902
40.8801	14.2978	58.8289	14.5229	18.6635	6.7436	52.8423	27.2880
39.9494	29.5114	47.5099	24.0664	10.1121	27.2662	28.7812	27.6659
8.0831	27.6705	9.1556	14.1304	53.7989	0.2199	33.6490	0.3980
1.3496	16.8359	49.9816	6.0828	19.3635	17.6622	36.9545	23.0265
15.7320	19.5697	11.5118	17.3884	44.0398	16.2635	39.7139	28.4203
6.9909	23.1804	38.3392	19.9950	24.6543	19.6057	36.9980	24.3992
4.1591	3.1853	40.1400	20.3030	23.9876	9.4030	41.1084	27.7149

13.3 模拟退火算法

这是一个旅行商问题。我们依次给基地编号为 1，敌方目标依次编号为 2, 3, ..., 101, 最后我方基地再重复编号为 102(这样便于程序中计算)。距离矩阵 $D = (d_{ij})_{102 \times 102}$ ，其中 d_{ij} 表示表示 i, j 两点的距离， $i, j = 1, 2, \dots, 102$ ，这里 D 为实对称矩阵。则问题是求一个从点 1 出发，走遍所有中间点，到达点 102 的一个最短路径。

13.3 模拟退火算法

设A, B 两点的地理坐标分别为 (x_1, y_1) , (x_2, y_2) , 过 A, B 两点的大圆的劣弧长即为两点的实际距离。以地心为坐标原点 O , 以赤道平面为 XOY 平面, 以 0 度经线圈所在的平面为 XOZ 平面建立三维直角坐标系。则 A, B 两点的直角坐标分别为:

$$A(R \cos x_1 \cos y_1, R \sin x_1 \cos y_1, R \sin y_1)$$

$$B(R \cos x_2 \cos y_2, R \sin x_2 \cos y_2, R \sin y_2)$$

其中 $R = 6370$ 为地球半径。

A, B 两点的实际距离

$$d = R \arccos \left(\frac{\overrightarrow{OA} \cdot \overrightarrow{OB}}{|\overrightarrow{OA}| \cdot |\overrightarrow{OB}|} \right),$$

化简得

$$d = R \arccos [\cos(x_1 - x_2) \cos y_1 \cos y_2 + \sin y_1 \sin y_2]。$$

13.3 模拟退火算法

求解的模拟退火算法描述如下：

(1) 解空间

解空间 S 可表为 $\{1, 2, \dots, 101, 102\}$ 的所有固定起点和终点的循环排列集合，即

$$S = \{(\pi_1, \dots, \pi_{102}) \mid \pi_1 = 1, (\pi_2, \dots, \pi_{101}) \text{ 为 } \{2, 3, \dots, 101\} \text{ 的循环排列}, \pi_{102} = 102\}$$

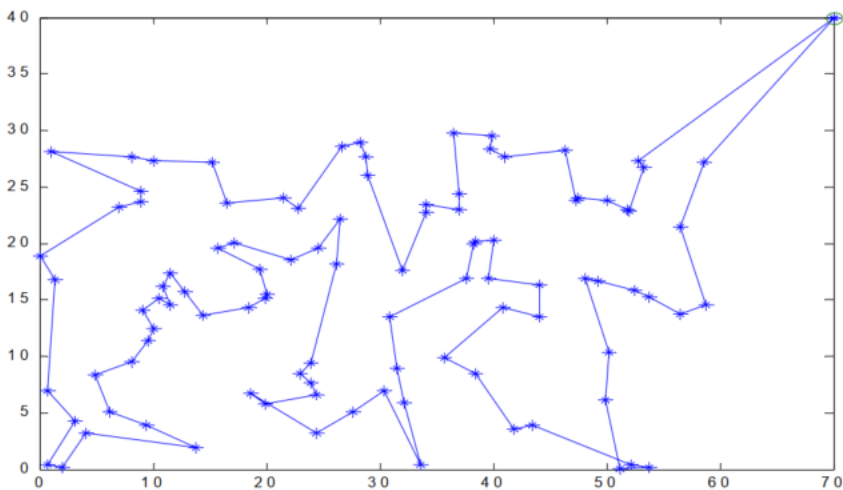
其中每一个循环排列表示侦察 100 个目标的一个回路， $\pi_i = j$ 表示在第 $i-1$ 次侦察目标 j ，初始解可选为 $(1, 2, \dots, 102)$ ，本文中我们先使用 Monte Carlo 方法求得一个较好的初始解。

(2) 目标函数

此时的目标函数为侦察所有目标的路径长度或称代价函数。我们要求

$$\min f(\pi_1, \pi_2, \dots, \pi_{102}) = \sum_{i=1}^{101} d_{\pi_i \pi_{i+1}}$$

13.3 模拟退火算法



(3) 新解的产生

① 2 变换法

任选序号 u, v ($u < v$) 交换 u 与 v 之间的顺序, 此时的新路径为:

$$\pi_1 \cdots \pi_{u-1} \pi_v \pi_{v-1} \cdots \pi_{u+1} \pi_u \pi_{v+1} \cdots \pi_{102}$$

② 3 变换法

任选序号 u, v 和 w , 将 u 和 v 之间的路径插到 w 之后, 对应的新路径为 (设 $u < v < w$)

$$\pi_1 \cdots \pi_{u-1} \pi_{v+1} \cdots \pi_w \pi_u \cdots \pi_v \pi_{w+1} \cdots \pi_{102}$$

(4) 代价函数差

对于 2 变换法, 路径差可表示为

$$\Delta f = (d_{\pi_{u-1}\pi_v} + d_{\pi_u\pi_{v+1}}) - (d_{\pi_{u-1}\pi_u} + d_{\pi_v\pi_{v+1}})$$

(5) 接受准则

$$P = \begin{cases} 1 & \Delta f < 0 \\ \exp(-\Delta f / T) & \Delta f \geq 0 \end{cases}$$

如果 $\Delta f < 0$, 则接受新的路径。否则, 以概率 $\exp(-\Delta f / T)$ 接受新的路径, 即若 $\exp(-\Delta f / T)$ 大于 0 到 1 之间的随机数则接受。

(6) 降温

利用选定的降温系数 α 进行降温即: $T \leftarrow \alpha T$, 得到新的温度, 这里我们取 $\alpha = 0.999$ 。

(7) 结束条件

用选定的终止温度 $e = 10^{-30}$, 判断退火过程是否结束。若 $T < e$, 算法结束, 输出当前状态。

13.3 模拟退火算法

```
clc,clear
load sj.txt %加载敌方 100 个目标的数据, 数据按照表格中的位置保存在
纯文本
文件 sj.txt 中
x=sj(:,1:2:8);x=x(:);
y=sj(:,2:2:8);y=y(:);
sj=[x y]; d1=[70,40];
sj=[d1;sj;d1]; sj=sj*pi/180;
d=zeros(102); %距离矩阵 d
for i=1:101
    for j=i+1:102
        temp=cos(sj(i,1)-sj(j,1))*cos(sj(i,2))*cos(sj(j,2))+sin(sj(i,2))*sin(sj(j,2));
        d(i,j)=6370*acos(temp);
    end
end
d=d+d';
S0=[];Sum=inf;
rand('state',sum(clock));
for j=1:1000
    S=[1 1+randperm(100),102];
    temp=0;
    for i=1:101
        temp=temp+d(S(i),S(i+1));
    end
    if temp<Sum
        S0=S;Sum=temp;
    end
end
e=0.1^30;L=20000;at=0.999;T=1;
```

```
%退火过程
for k=1:L
    %产生新解
    c=2+floor(100*rand(1,2));
    c=sort(c);
    c1=c(1);c2=c(2);
    %计算代价函数值
    df=d(S0(c1-1),S0(c2))+d(S0(c1),S0(c2+1))-d(S0(c1-1),S0(c1))-d(S0(c2),S0(c2+1));
    %接受准则
    if df<0
        S0=[S0(1:c1-1),S0(c2:-1:c1),S0(c2+1:102)];
        Sum=Sum+df;
    elseif exp(-df/T)>rand(1)
        S0=[S0(1:c1-1),S0(c2:-1:c1),S0(c2+1:102)];
        Sum=Sum+df;
    end
    T=T*at;
    if T<e
        break;
    end
end
% 输出巡航路径及路径长度
S0,Sum
```

13.4 遗传算法

遗传算法的生物学基础

生物在自然界中的生存繁衍，显示出了其对自然环境的自适应能力。受其启发，人们致力于对生物各种生存特性的机理研究和行为模拟，为人工自适应系统的设计和开发提供了广阔的前景。

遗传算法(Genetic Algorithms, 简称GAs)就是这种生物行为的计算机模拟中令人瞩目的重要成果。

13.4 遗传算法

遗传算法是模拟生物在自然环境下的遗传和进化过程而形成的一种自适应全局优化概率搜索方法。

它最早由美国密西根大学的H.Holland教授提出，起源于60年代对自然和人工自适应系统的研究；

1967年，Bagley发表了关于遗传算法应用的论文，在其论文中首次使用“遗传算法（Genetic Algorithm）”一词。

70年代 De Jong基于遗传算法的思想在计算机上进行了大量的纯数值函数优化计算实验。

在一系列研究工作的基础上，80年代由Goldberg进行归纳总结，形成了遗传算法的基本框架。

13.4 遗传算法

遗传(Heredity)——世间的生物从其父代继承特性或性状，这种生命现象就称为遗传(Heredity)

遗传信息是由**基因(Gene)**组成的，生物的各种性状由其相应的基因所控制。

- **基因是遗传的基本单位**。细胞通过分裂具有自我复制的能力，在细胞分裂的过程中，其**遗传基因**也同时被复制到下一代，从而其性状也被下一代所继承。

13.4 遗传算法

生物的遗传方式

1. 复制

生物的主要遗传方式是复制。遗传过程中，父代的遗传物质DNA被复制到子代。即细胞在分裂时，遗传物质DNA通过复制(Reproduction)而转移到新生的细胞中，新细胞就继承了旧细胞的基因。

2. 交叉

有性生殖生物在繁殖下一代时，两个同源染色体之间通过交叉(Crossover)而重组，亦即在两个染色体的某一相同位置处DNA被切断，其前后两串分别交叉组合而形成两个新的染色体。

3. 变异

在进行细胞复制时，虽然概率很小，仅仅有可能产生某些复制差错，从而使DNA发生某种变异(Mutation)，产生出新的染色体。这些新的染色体表现出新的性状。

如此这般，遗传基因或染色体在遗传的过程中由于各种各样的原因而发生变化。

13.4 遗传算法

将 n 维决策向量 X 用 n 个记号 $x_i (i = 1, 2, \dots, n)$ 所组成的符号串 X 来表示:

$$X = [x_1, x_2, \dots, x_n]^T$$

- 把每一个 x_i 看作一个遗传基因，这样， X 就可看做是由 n 个遗传基因所组成的一个染色体。
- 这里的等位基因可以是一组整数。也可以是某一范围内的实数值，或者是纯粹的一个记号。
最简单的等位基因是由0和1这两个整数组成的，相应的染色体就可表示为一个二进制符号串。

13.4 遗传算法

- 这种编码所形成的排列形式 X 是个体的**基因型**，与它对应的 X 值是个体的**表现型**。
- 对于每一个个体 X ，要按照一定的规则确定出其**适应度**，个体的适应度与其对应的个体表现型 X 的目标函数值相关联， X 越接近于目标函数的最优点，其适应度越大；反之，其适应度越小。

遗传算法中，决策变量 X 组成了问题的解空间。对问题最优解的搜索是通过对染色体 X 的搜索过程来进行的。从而所有的染色体 X 就组成了问题的搜索空间。

13.4 遗传算法

生物的进化是以集团为主体的。与此相对应，遗传算法的运算对象是由M个个体所组成的集合，称为**群体（或称种群）**。与生物一代一代的自然进化过程相类似，遗传算法的运算过程也是一个反复迭代过程：

第 t 代群体记做 $P(t)$ ，经过一代遗传和进化后，得到 $t + 1$ 代群体，记做 $P(t + 1)$ 。

这个群体不断地经过遗传和进化操作，并且每次都按照优胜劣汰的规则将适应度较高的个体更多地遗传到下一代，这样最终在群体中将会得到一个优良的个体 X ，它所对应的表现型 X 将达到或接近于问题的最优解 X^* 。

13.4 遗传算法

生物遗传概念在遗传算法中的对应关系

生物遗传概念	遗传算法中的作用
适者生存	算法停止时，最优目标值的解有最大的可能被留住
个体	解
染色体	解的编码
基因	解中每一分量的特征
适应性	适应度函数值
种群	根据适应度函数值选取的一组解
交配	通过交配原则产生一组新解的过程
变异	编码的某一分量发生变化的过程

13.4 遗传算法

选择(复制):

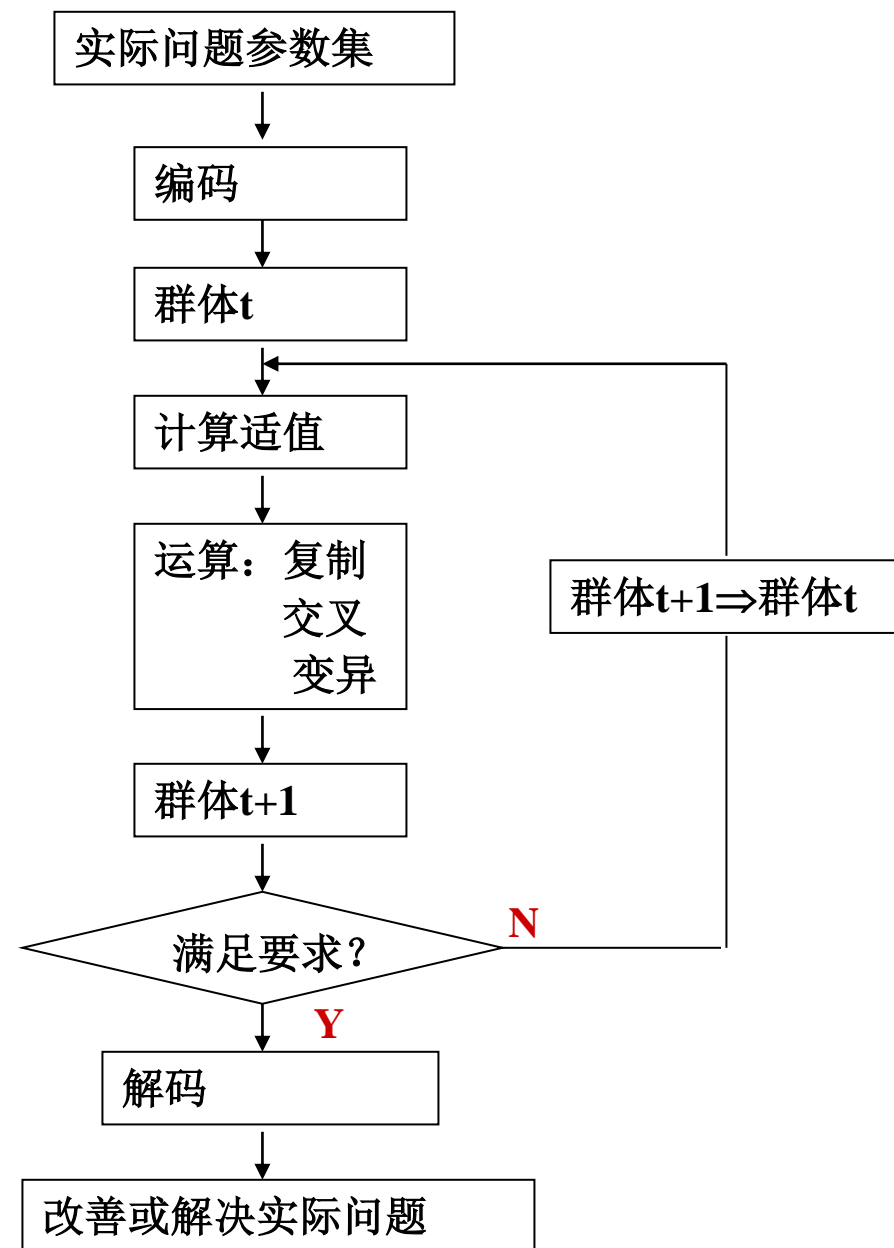
根据各个个体的适应度, 按照一定的规则或方法, 从第 t 代群体 $P(t)$ 中选择出一些优良的个体遗传到下一代群体 $P(t+1)$ 中;

交叉:

将群体 $P(t)$ 内的各个个体随机搭配成对, 对每一对个体, 以某个概率(称为交叉概率) 交换它们之间的部分染色体;

变异:

对群体 $P(t)$ 中的每一个个体, 以某一概率(称为变异概率)改变某一个或某一些基因座上的基因值为其他基因值。



13.4 遗传算法

遗传算法的应用

遗传算法提供了一种求解复杂系统优化问题的通用框架，它不依赖于问题的具体领域，对问题的种类有很强的鲁棒性，所以广泛应用于很多学科。下面是遗传算法的一些主要应用领域：

(1) 函数优化

函数优化是遗传算法的经典应用领域，也是对遗传算法进行性能评价的常用算例。

对于一些非线性、多模型、多目标的函数优化问题，用其他优化方法较难求解，用遗传算法可以方便地得到较好的结果。

(2) 组合优化

随着问题规模的增大，组合优化问题的搜索空间也急剧扩大，有时在目前的计算机上用枚举法很难或甚至不可能求出其精确最优解。对这类复杂问题，人们已意识到应把主要精力放在寻求其满意解上，而遗传算法是寻求这种满意解的最佳工具之一。实践证明，遗传算法对于组合优化中的NP完全问题非常有效。

例如，遗传算法已经在求解旅行商问题、背包问题、装箱问题、图形划分问题等方面得到成功的应用。

13.4 遗传算法

(3) 生产调度问题

生产调度问题在很多情况下所建立起来的数学模型难以精确求解，即使经过一些简化之后可以进行求解，也会因简化得太多而使得求解结果与实际相差甚远。而目前在现实生产中也主要是靠一些经验来进行调度。

现在遗传算法已成为解决复杂调度问题的有效工具，在单件生产车间调度、流水线生产车间调度、生产规划、任务分配等方面遗传算法都得到了有效的应用。

(4) 自动控制

在自动控制领域中很多与优化相关的问题需要求解，遗传算法已在其中得到了初步的应用，并显示出了良好的效果。

例如用遗传算法进行航空控制系统的优化、使用遗传算法设计空间交会控制器、基于遗传算法的模糊控制器的优化设计、基于遗传算法的参数辨识、基于遗传算法的模糊控制规则的学习、利用遗传算法进行人工神经网络的结构优化设计和权值学习等，都显示出了遗传算法在这些领域中应用的可能性。

13.4 遗传算法

(5) 机器人学

机器人是一类复杂的难以精确建模的人工系统，而遗传算法的起源就来自于对人工自适应系统的研究，所以机器人学理所当然地成为遗传算法的一个重要应用领域。

例如，遗传算法已经在移动机器人路径规划、关节机器人运动轨迹规划、机器人逆运动学求解、细胞机器人的结构优化和行为协调等方面得到研究和应用。

(6) 图像处理

图像处理是计算机视觉中的一个重要研究领域。在图像处理过程中，如扫描、特征提取、图像分割等不可避免地会存在一些误差，这些误差会影响图像处理的效果。如何使这些误差最小是使计算机视觉达到实用化的重要要求。

遗传算法在这些图像处理中的优化计算方面找到了用武之地，日前已在模式识别、图像恢复、图像边缘特征提取等方面得到了应用。

(7) 人工生命

人工生命是用计算机、机械等人工媒体模拟或构造出的具有自然生物系统特有行为的人造系统。自组织能力和自学习能力是人工生命的两大主要特征。人工生命与遗传算法有着密切的关系，基于遗传算法的进化模型是研究人工生命现象的重要基础理论。

13.4 遗传算法

求解13.3中TSP问题：

求解的遗传算法的参数设定如下：

种群大小： $M = 50$

最大代数： $G = 1000$

交叉率： $p_c = 1$ ，交叉概率为 1 能保证种群的充分进化。

变异率： $p_m = 0.1$ ，一般而言，变异发生的可能性较小。

(1) 编码策略

采用十进制编码，用随机数列 $\omega_1\omega_2\ldots\omega_{102}$ 作为染色体，其中 $0 < \omega_i < 1$ ($i = 2, 3, \dots, 101$)， $\omega_1 = 0$ ， $\omega_{102} = 1$ ；每一个随机序列都和种群中的一个个体相对应，例如一个 9 目标问题的一个染色体为

[0.23, 0.82, 0.45, 0.74, 0.87, 0.11, 0.56, 0.69, 0.78]

其中编码位置 i 代表目标 i ，位置 i 的随机数表示目标 i 在巡回中的顺序，我们将这些随机数按升序排列得到如下巡回：

6—1—3—7—8—4—9—2—5

13.4 遗传算法

(2) 初始种群

本文中我们先利用经典的近似算法—改良圈算法求得一个较好的初始种群。即对于初始圈

$C = \pi_1 \cdots \pi_{u-1} \pi_u \pi_{u+1} \cdots \pi_{v-1} \pi_v \pi_{v+1} \cdots \pi_{102}$, $2 \leq u < v \leq 101$, $2 \leq \pi_u < \pi_v \leq 101$, 交换 u 与 v 之间的顺序, 此时的新路径为:

$$\pi_1 \cdots \pi_{u-1} \pi_v \pi_{v-1} \cdots \pi_{u+1} \pi_u \pi_{v+1} \cdots \pi_{102}$$

记 $\Delta f = (d_{\pi_{u-1}\pi_v} + d_{\pi_u\pi_{v+1}}) - (d_{\pi_{u-1}\pi_u} + d_{\pi_v\pi_{v+1}})$, 若 $\Delta f < 0$, 则以新的路径修改旧的路径, 直到不能修改为止。

(3) 目标函数

目标函数为侦察所有目标的路径长度, 适应度函数就取为目标函数。我们要求

$$\min f(\pi_1, \pi_2, \cdots, \pi_{102}) = \sum_{i=1}^{101} d_{\pi_i \pi_{i+1}}$$

13.4 遗传算法

https://www.zhihu.com/tardis/zm/art/158600868?source_id=1003

(4) 交叉操作

我们的交叉操作采用单点交叉。设计如下，对于选定的两个父代个体 $f_1 = \omega_1 \omega_2 \dots \omega_{102}$ ， $f_2 = \omega'_1 \omega'_2 \dots \omega'_{102}$ ，我们随机地选取第 t 个基因处为交叉点，则经过交叉运算后得到的子代编码为 s_1 和 s_2 ， s_1 的基因由 f_1 的前 t 个基因和 f_2 的后 $102 - t$ 个基因构成， s_2 的基因由 f_2 的前 t 个基因和 f_1 的后 $102 - t$ 个基因构成，例如：

$$f_1 = [0, 0.14, 0.25, 0.27, | 0.29, 0.54, \dots, 0.19, 1]$$

$$f_2 = [0, 0.23, 0.44, 0.56, | 0.74, 0.21, \dots, 0.24, 1]$$

设交叉点为第四个基因处，则

$$s_1 = [0, 0.14, 0.25, 0.27, | 0.74, 0.21, \dots, 0.24, 1]$$

$$s_2 = [0, 0.23, 0.44, 0.56, | 0.29, 0.54, \dots, 0.19, 1]$$

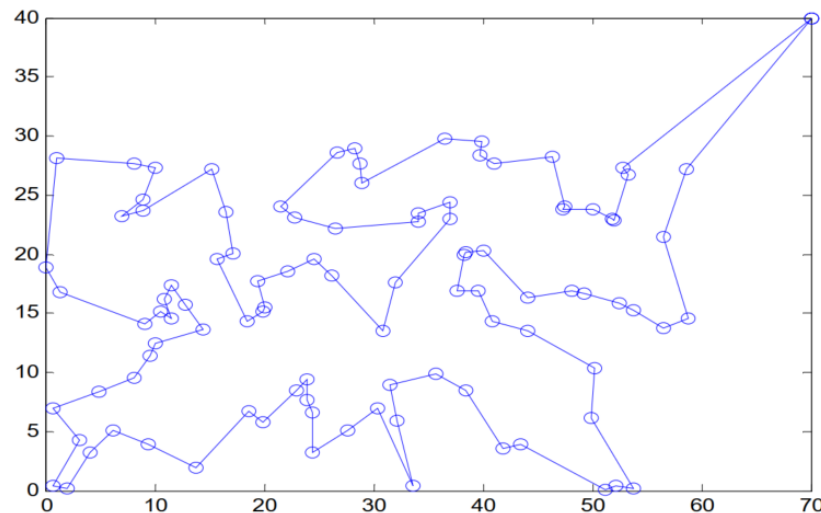
交叉操作的方式有很多种选择，我们应该尽可能选取好的交叉方式，保证子代能继承父代的优良特性。同时这里的交叉操作也蕴含了变异操作。

(5) 变异操作

变异也是实现群体多样性的一种手段，同时也是全局寻优的保证。具体设计如下，按照给定的变异率，对选定变异的个体，随机地取三个整数，满足 $1 < u < v < w < 102$ ，把 u, v 之间（包括 u 和 v ）的基因段插到 w 后面。

(6) 选择

采用确定性的选择策略，也就是说选择目标函数值最小的 M 个个体进化到下一代，这样可以保证父代的优良特性被保存下来。



13.4 遗传算法

```
tic
clc,clear
load sj.txt %加载敌方 100 个目标的数据
x=sj(:,1:2:8); x=x(:);
y=sj(:,2:2:8); y=y(:);
sj=[x y]; d1=[70,40];
sj0=[d1,sj;d1]; sj=sj0*pi/180;
d=zeros(102); %距离矩阵 d
for i=1:101
    for j=i+1:102
        temp=cos(sj(i,1)-
            sj(j,1))*cos(sj(i,2))*cos(sj(j,2))+sin(sj(i,2))*sin(sj(j,2));
        d(i,j)=6370*acos(temp);
    end
end
d=d+d';L=102;w=50;dai=100;
%通过改良圈算法选取优良父代 A
for k=1:w
    c=randperm(100);
    c1=[1,c+1,102];
    flag=1;
```

```
while flag>0
    flag=0;
    for m=1:L-3
        for n=m+2:L-1
            if
                d(c1(m),c1(n))+d(c1(m+1),c1(n+1))<d(c1(m),c1(m
                    +1))+d(c1(n),c1(n+1))
                flag=1;
                c1(m+1:n)=c1(n:-1:m+1);
            end
        end
    end
    J(k,c1)=1:102;
end
J=J/102;
J(:,1)=0;J(:,102)=1;
rand('state',sum(clock));
%遗传算法实现过程
A=J;
for k=1:dai %产生 0 ~ 1 间随机数列进行编码
    B=A;
    c=randperm(w);
    %交配产生子代 B
    for i=1:2:w
        F=2+floor(100*rand(1));
        temp=B(c(i),F:102);
        B(c(i),F:102)=B(c(i+1),F:102);
        B(c(i+1),F:102)=temp;
    end
    %变异产生子代 C
    by=find(rand(1,w)<0.1);
    if length(by)==0
        by=floor(w*rand(1))+1;
    end
```

```
C=A(by,:);
L3=length(by);
for j=1:L3
    bw=2+floor(100*rand(1,3));
    bw=sort(bw);
    C(j,:)=C(j,[1:bw(1)-
        1,bw(2)+1:bw(3),bw(1):bw(2),bw(3)+1:102]);
end
G=[A;B;C];
TL=size(G,1);
%在父代和子代中选择优良品种作为新的父代
[dd,IX]=sort(G,2);temp(1:TL)=0;
for j=1:TL
    for i=1:101
        temp(j)=temp(j)+d(IX(j,i),IX(j,i+1));
    end
end
[DZ,IZ]=sort(temp);
A=G(IZ(1:w),:);
end
path=IX(IZ(1),:)
long=DZ(1)
toc
xx=sj0(path,1);yy=sj0(path,2);
plot(xx,yy,'-o')
```

总结

现代优化算法

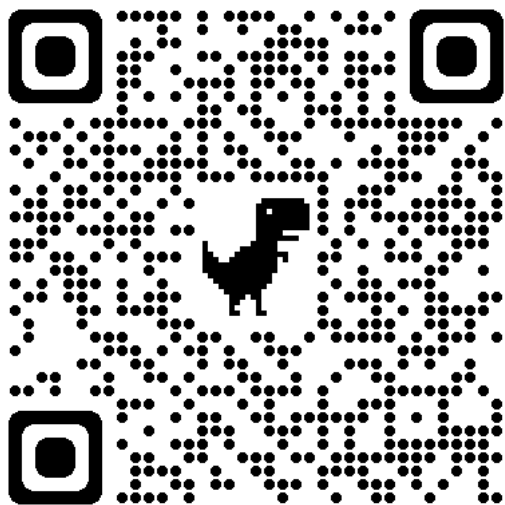
禁忌搜索算法

模拟退火算法

遗传算法



Q&A?



翟晓雅

Email: xiaoyazhai@ustc.edu.cn

Homepage: <https://xiaoyazhai.github.io/>

Lab: <http://gcl.ustc.edu.cn/>