遗传算法求解TSP问题

孟丽媛　3180101844

2020.11

1　简单遗传算法求解TSP问题

1.1　问题简述

TSP（Traveling Salesman Problem）问题：给定n 个城市和每两个城市间的距离（无向完全图，有权边），选择一条线路，使得每个城市只经过一次且仅有一次，并最终回到起点。求能完成这个遍历的总路径长度最短。

对于完全图，我们用邻接矩阵（adjacency matrix）$D = [d_{ij}]$ 来表示全部城市，其中d_{ij} 表示城市i和j 之间的距离。显然，d_{ij}满足：$d_{ij} \geq 0, 1 \leq i, j \leq n$; $d_{ii} = 0, 1 \leq i \leq n$; $d_{ij} = d_{ji}, 1 \leq i, j \leq n$; $d_{ij} + d_{jk} \geq d_{ik}, 1 \leq i, j, k \leq n$.

而问题的一个解可以表达为一个循环排列

$$\pi = (\pi_1, \pi_2, \cdots, \pi_n),$$

表示路径

$$\pi_1 \to \pi_2 \to \cdots \to \pi_n,$$

其中$\pi_i \in \{1, 2, \cdots, n\}, i = 1, 2, \cdots, n$ 表示各个城市的编号，显然对可行解，有$i \neq j$，则$\pi_i \neq \pi_j$。

1.2　算法设计思路

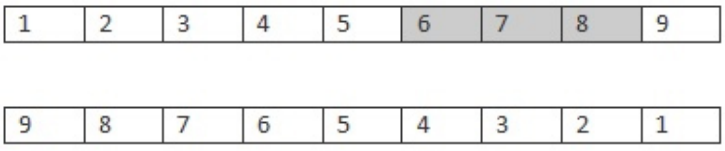
TSP 问题的解本身就是一组不重复的唯一编码：

$$i_1 \to i_2 \to \cdots \to i_l.$$

这里直接采用此编码为DNA[1]。

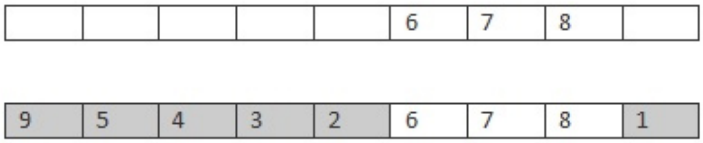
TSP问题的优化目标是距离最短，适应度取为总距离的倒数。按适应度的大小抽取亲代个体产生下一代的新个体。产生新个体的过程中将按aCrossRate，aMutationRage的概率发生交叉和突变（交叉和突变不会同时发生）。同时每产生新的一代时，都保留上一代适应度最优的个体以保证迭代过程收敛[1]。给定一个参数aLifeCount，迭代aLifeCount代以后就终止。

交叉算子：
按适应度随机选择亲代的两个个体用parent1和parent2表示。当发生了染色体交叉时将随机选取Parent1中连续的一部分基因替换掉parent2对应位置的基因。例如，随机选取parent1的基因片段678：



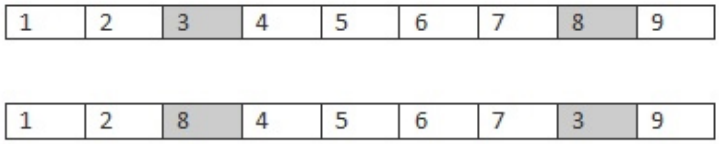
[1]不增加这一条的情况下，测试了多组交叉和突变的参数，即使当种群很大，迭代次数很多时，遗传算法也还是不收敛。

与parent2交叉就得到一个子代:



突变算子:

在TSP问题中因为每个城市只经过一次，所以在变异的时候不能只是改变基因序列中的某一位的值（这会导致一个城市经过两次），应该随机交换两个位置的值，如:



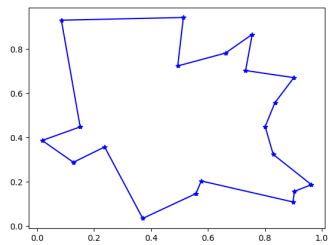
1.3 测试结果与结果分析

分别选取20,30,50,100,200个城市，测试多组交叉和突变参数，调整种群大小和迭代次数，使得迭代结果稳定，记录此结果。

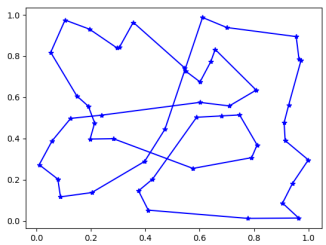
Table 1: 简单遗传算法求解TSP问题

城市数	N=20	N=30	N=50	N=100	N=200
交叉概率	0.7	0.7	0.7	0.7	0.7
突变概率	0.3	0.3	0.3	0.3	0.8
种群大小	100	100	100	100	100
迭代稳定时的迭代次数	800	2300	12000	50000	100000
总环游路线长	3.842867	5.155315	7.464181	9.785882	19.180739

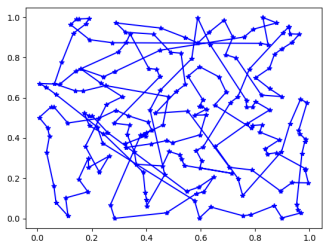
画出城市数为20,50,200时的优化后环游路线图:



(a) 城市数20



(b) 城市数50



(c) 城市数200

因为每一代总是会保留上一代的最优个体，所以可以适当增大交叉和突变的概率来减小所需的迭代次数而不会使得算法不收敛，较大的交叉和突变的概率也更低可能地会陷入局部最优解。随着城市数的增多，选取的种群规模也应当增大，以保证迭代的种群中可以保有足够多样的基因库[2]。

[2]此次实验考虑运行时间的因素故没有调整种群大小。

但从最终优化结果的环游路线图发现当城市数较多时即使交叉和突变的概率很高，迭代次数很多，也无法得到最优环游。

1.4 遗传算法与模拟退火算法的比较

分别选取20,30,50,100,200个城市，记录遗传算法与模拟退火算法的测试结果。

Table 2: 简单遗传算法与模拟退火法比较

城市数	N=20	N=30	N=50	N=100	N=200
简单遗传算法总环游路线长	3.842867	5.155315	7.464181	9.785882	19.180739
模拟退火法总环游路线长	3.9321	5.0305	5.7500	8.2093	11.4386

在最终优化效果和计算时间两个角度上，模拟退火法均明显优于简单遗传算法。观察最终结果的环游路线图，模拟退火法的优化结果就是最优环游，而简单遗传算法的优化结果当城市数较多时常常无法得到最优环游。

2 改进的遗传算法求解TSP问题

在进行简单遗传算法的测试时发现算法收敛的很慢，同一参数不同次测试的优化结果相差较大，最终解不是问题的最优解。

2.1 算法改进思路

在原有算法中选择变化幅度更大的适应度函数(这里选为总距离三次方的倒数），这样可以增大不同个体之间的差异，让种群进化得更快，同时也可以让优质基因更大概率地保留下来，改进优化结果。

2.2 测试结果与结果分析

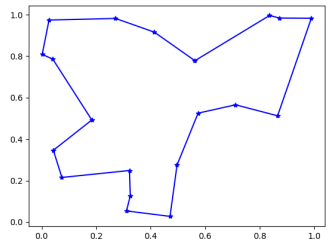
分别选取20,30,50,100,200个城市，测试多组交叉和突变参数，调整种群大小和迭代次数，使得迭代结果稳定，记录此结果。

Table 3: 改进的遗传算法求解TSP问题

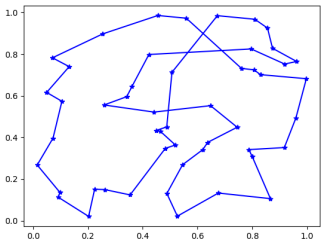
城市数	N=20	N=30	N=50	N=100	N=200
交叉概率	0.7	0.7	0.7	0.7	0.7
突变概率	0.3	0.3	0.3	0.3	0.8
种群大小	100	100	100	100	100
迭代稳定时的迭代次数	170	1300	4700	47000	98000
总环游路线长	3.898243	4.774596	6.777371	9.940915	17.206587

与简单遗传算法的结果相比较，改进后的遗传算法能更快地收敛，在运算时间上表现更好，最终的优化结果也更好。

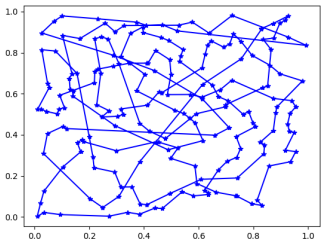
画出城市数为20,50,200时的优化后环游路线图:



(d) 城市数20



(e) 城市数50



(f) 城市数200

环游路线图也体现出与简单的遗传算法相比改进后的遗传算法更优化结果更好。

3 由遗传算法和模拟退火算法构建新算法简述

先初始化一个随机种群，由遗传算法产生新一代个体，然后对这一代个体进行模拟退火，将模拟退火后的这一代个体作为遗传算法过程中产生的新种群，重复此过程直至满足优化要求。[2]

将二者结合的新算法可以兼具遗传算法的全局搜索功能和模拟退火法的局部搜索功能，而这样的代价是运算时间大大增加了，这一算法实际上的运行效果还需要进一步的测试来评价。

4 总结

在与同学交流后，发现我的简单遗传算法程序计算时间较长，所需迭代次数更多，但因为时间原因就没有对程序做更多地调整，所以这个报告里的测试结果并不能很好的反映出算法在运算时间方面的优劣，程序编写的优劣可能对算法所需运算时间造成了很大影响。另外，这个报告中的模拟退火算法程序用的是老师上课的Matlab程序，而简单模拟退火算法程序用的是Python程序，编译器的不同可能也为运算时间的差异带来了一定影响。

参考文献

[1] 余有明,刘玉树,阎光伟.遗传算法的编码理论与应用[J].计算机工程与应用,2006(03):86-89.

[2] 王银年. 遗传算法的研究与应用[D].江南大学,2009.