遗传算法求解TSP问题

**摘要：**

TSP问题是指假设有一个旅行商人要拜访n个城市，他必须选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。本文使用遗传算法解决att48问题，即48个城市的旅行商问题，该问题目前的最优解是10628，受个别参数影响以及数据样本数量的限制，本文所得到的最优结果是10648，相对误差为0.18818216%。

**一、导言**

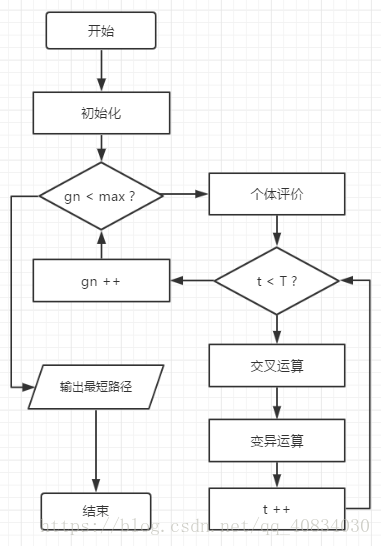
旅行商问题是一个经典的组合优化问题。一个经典的旅行商问题可以描述为：一个商品推销员要去若干个城市推销商品，该推销员从一个城市出发，需要经过所有城市后，回到出发地。应如何选择行进路线，以使总的行程最短。从图论的角度来看，该问题实质是在一个带权完全无向图中，找一个权值最小的Hamilton回路。由于该问题的可行解是所有顶点的全排列，随着顶点数的增加，会产生组合爆炸，它是一个NP完全问题。

TSP问题可以分为对称和不对称。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，而不对称TSP则形成有向图。对称性TSP问题可以将解的数量减少了一半。所以本次实验的TSP问题使用att48数据，可在tsplib中下载数据包。

演化算法是一类模拟自然界遗传进化规律的仿生学算法，它不是一个具体的算法，而是一个算法簇。遗传算法是演化算法的一个分支，由于遗传算法的整体搜索策略和优化计算是不依赖梯度信息，所以它的应用比较广泛。我们本次实验同样用到了遗传算法（用MATLAB编写）来解决TSP问题。

**二、算法流程**

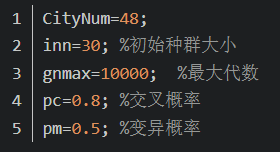
遗传算法的基本流程图如下图所示。gn对应当前代数，max为最大代数，T种群大小。



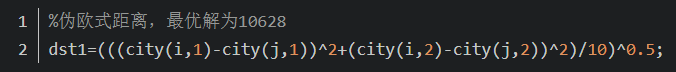
基于流程图，分析一下我们的算法实现代码：

**2.1 初始化**

我们初始化以下常量，由于我们要解决的是att48旅行商问题，所以城市数量是48。



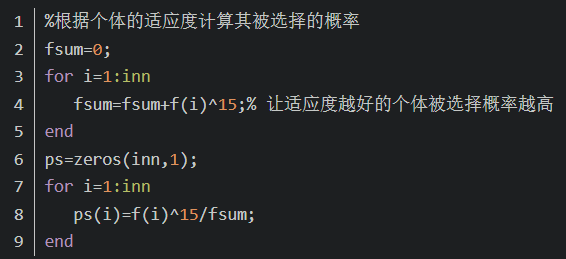
然后对这48个城市进行整数编码（1 - 48），并根据提供的城市坐标计算每两个城市之间的距离。由于两个城市之间的实际距离比较大，不方便计算，所以这里距离的计算方法是采用伪欧式距离方法，采用实际距离的10^0.5倍，并四舍五入，保留整数。



完成以上步骤之后，随机产生一个种群作为初始种群，同时计算这个初始种群的适应度。

**2.2 个体评价**

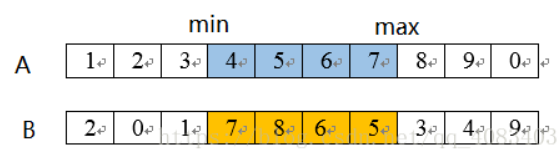
首先我们需要计算每个个体的适应度，适应度越高，被保留概率越大。我们取总距离的倒数作为适应度。为了增大适应度高的个体的被选中概率，我们利用以下公式计算个体被选择的概率。注意每次完成交叉、变异运算之后需要重新评价。



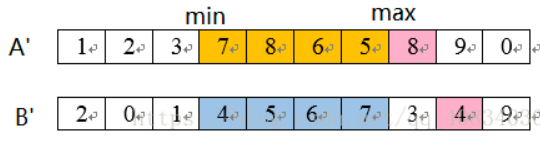
**2.3 交叉运算**

选择两个个体进行交叉操作。首先在[1，sz]（sz为城市数数量）的范围内，即在染色体长度内，随机产生两个交叉位min和max（min<max），将两个个体的[min，max]区域互换，示意图如下图所示。

(a) 交叉前：



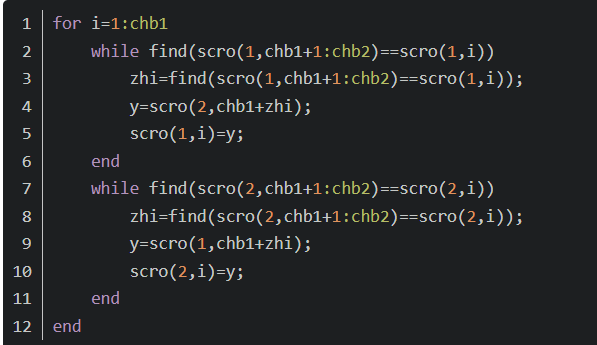
(b) 交叉后：



根据上图，A、B是未发生交叉之前的染色体（个体），A’和B’是A、B发生交叉之后产生的新个体。通过冲突检测，我们可以发现交叉之后会产生同一个基因在同一条染色体上可能会重复出现，这就是交叉的冲突(图(b)中粉色标注部分)。解决交叉冲突的方法如下，

（1）先检测[1，min]区域的基因是否和[min，max]区域中的基因冲突，如果有冲突，那么进行一下操作来消除冲突。

检测染色体A的左边区域[1，min]是否与染色体A的交叉区域[min，max]存在基因重复，如果有，那么记录染色体A的交叉区域中，重复基因的位置p，左边区域中的重复基因位置为p\_left，然后将染色体B中min+p位置的基因复制给染色体A中p\_left位置。重复该步骤直到染色体A的[1，min]区域的基因和[min，max]区域中没有基因冲突。对染色体B进行同样的操作。具体实现代码如下所示。其中，变量chb1、chb2分别对应min和max，zhi对应p。

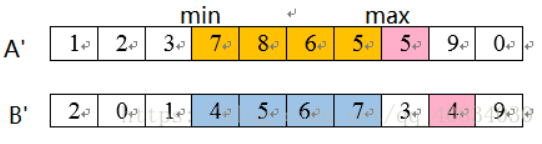


（2）检测[max，END]区域的基因是否和[min，max]区域中的基因冲突，如果有冲突，那么进行一下操作来消除冲突。

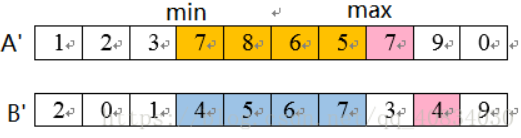
检测染色体A的右边区域[max，END]是否与染色体A的交叉区域[min，max]存在基因重复，如果有，那么记录染色体A的交叉区域中，重复基因的位置p，右边区域中的重复基因位置为p\_right，然后将染色体B中p位置的基因复制给染色体A中p\_right位置。重复该步骤直到染色体A的[max，END]区域的基因和[min，max]区域中没有基因冲突。对染色体B进行同样的操作。

对图（b）消除基因冲突的示意图如下图所示。

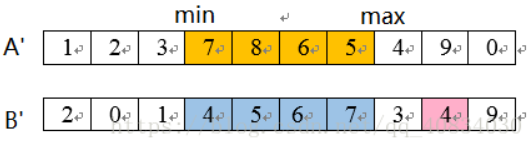
(c)将染色体A冲突基因8改为染色体B对应位置的5



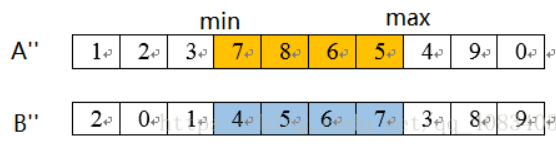
(d)将染色体A冲突基因5改为染色体B对应位置的7



(e)将染色体A冲突基因7改为染色体B对应位置的4



(f)对染色体B同样进行消除冲突



**2.4 变异运算**

对个体进行变异。首先在随机产生两个变异位min、max，其中min<max，且0<min，max<染色体长度。然后将选中的变异区域反顺，对图(f)中染色体A进行变异的结果图如图所示。



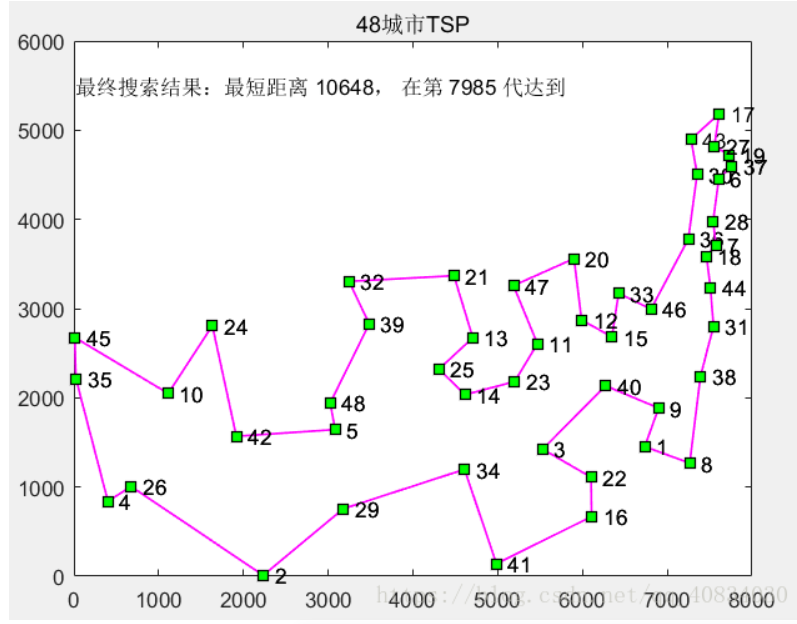
**2.5 输出结果**

我们设计的算法的终止条件是达到最大代数的迭代次数，每一次迭代结束后将得到的路径长度和当前代数（第几代）记录在数组中，然在搜索完成之后，将数组中记录的最短路径和对应的代数输出，作为我们的搜索结果。

**三、结果分析**

求解TSP问题是利用MATLAB语言编程的，解决TSP的演化算法采用遗传算法。由于遗传算法包含了随机搜索方法，所求的最优解不一定是最优的。在求解过程中，发现遗传算法得到的结果的精确度除了和交叉算子、变异算子、适应度计算方法有关，还受交叉概率、变异概率、迭代次数的影响。

对于本文算法，在一定范围内，迭代次数也大，变异概率越小，遗传算法的精确度越高；执行时间随着迭代次数的增加而增加。当交叉概率为0.8，变异概率为0.5，最大代数为10000时，能得到较理想的结果，如下图：



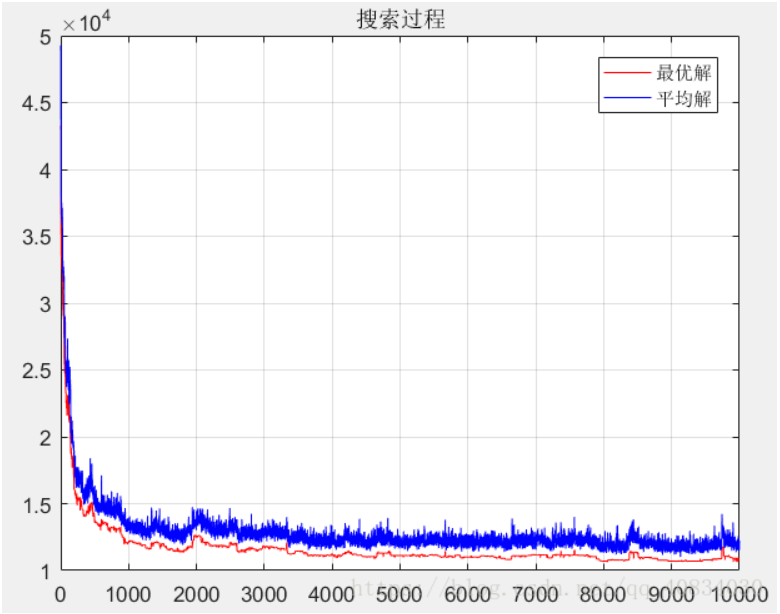
遗传算法的终止条件如下，满足其一即可终止算法。

**1 当最优个体的适应度达到给定的阈值；**

**2 最优个体的适应度和群体适应度不再上升；**

**3 迭代次数达到预设的代数时，算法终止。**

本文所设计的遗传算法的终止条件是3）条件。下图是当交叉概率为0.8，变异概率为0.5，最大代数为10000的最优解的搜索曲线，横轴代表迭代次数，纵轴表示搜索的距离。在图中我们可以发现，在搜索在1000代后，搜索过程曲线开始趋于平缓，达到4000代之后曲线基本不再变化，所以根据终止条件2），遗传算法是可以终止了，不必再继续计算到10000代，从而减少时间和资源的开销。因而我们的算法也可以从这个方面进行优化。



**四、结论**

由于遗传算法的整体搜索策略和优化计算是不依赖梯度信息，只需要影响搜索方向的目标函数和相应的适应度函数，所以它的应用比较广泛。利用遗传算来进行大规模问题的组合优化是一种比较有效的方法，因为在目前的计算上利用枚举法求解最优解有一定的难度。但是遗传算法也有不足之处，它对算法的精确度、可行度、计算复杂性等方面还没有有效的定量分析方法。通过本文的算法也可以清晰地认识到，遗传算法所求得的解不一定是最优解。