

遗传算法应用

学号：1004161221 姓名：白云鹏

一. 经典 TSP 问题

在使用 GA 求解 TSP 问题的时候，仍然需要先阐明染色体的设置，在 TSP 问题中，这是很显然的，我们以起始点作为染色体的开始以及结尾，染色体中每一个序列均为点的标号，这样一条染色体所代表的含义即为一条路径。每一条的路径直接模拟染色体进行遍历计算即可。

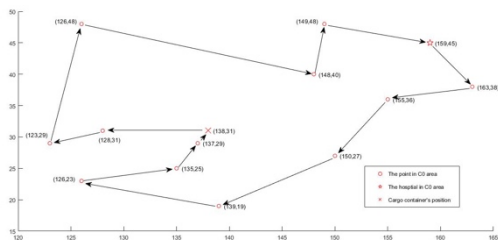
R1:

0	1	2	3	6	9	5	0
---	---	---	---	---	---	---	---

但是，在 TSP 问题中的变异以及交叉操作上，则出现了问题，在操作完毕之后，其路径可能会出现矛盾，即缺少点没有被访问，或者被重复访问，这时我们则需要重新设置，将路径有效。

在生成初始解时，我们随机的选取没有被遍历过的点，将其加入到现路径的末尾，直到全部的点都被访问过。我们就完成了初始解的设置。

最后，运行结果如下：在这里已经标出其路径。



用 GA 问题解决 TSP 问题，实际上只是对 GA 的一个简单入门了解，GA 的用途远不止解决这些经典问题，即使在现在，GA 仍体现出强大的生命力。

二. 在游戏中的简单应用

考虑这样一个场景，初始时，在坐标 $X = 0$ 处有一个点，每隔固定的时间间隔，它向 X 轴正向移动一个单位长度，在移动的过程中，它只有两个操作：

1. 向上移动 k_1 个单位长度
2. 向下移动 k_2 个单位长度

当然，在每一个 X 处都有一个允许通过的区域，如果其 Y 值不在该区域中，即停止。

这个简单的游戏场景就是 [Flappy Bird](#)，现在，我们则使用遗传算法来攻略这个游戏。

首先，我们设置其染色体组成，我们以操作 1 作为 True，操作 2 作为 False，那么每一次游戏的实际操作就是一段 2 进制序列：

L1:

1	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---

这样，我们对其染色体的操作就很清晰，其变异每次只变异一个点位。

那么，我们现在假设 $k_1 = 5, k_2 = 3$ ，长度为 100，高度为 20，来进行模拟，其结果如下：

