旅行商问题的遗传算法应用和分析

祝啸（21721270）

摘 要：遗传算法是一种生物启发式算法，借用生物遗传和变异的机制生成高质量的搜索和优化问题的解。本文介绍了遗传算法的基本机制，并以旅行商问题作为分析对象，对遗传算法的特性和不足进行了的展示和分析，并讨论了遗传算法变种的方式和变种的设计动机。

关键词：旅行商问题，遗传算法

1 引言

旅行商问题（TSP）是一个经典的组合优化问题。由于其问题形式简单，但又属于NP-hard问题，经常被用于启发式算法的测试和验证。

遗传算法（Genetic Algorithm）是一类基于自然界自然选择原理而设计的启发式算法，是进化算法（Evolutionary Algorithm）的一个分支。遗传算法采用“基因形式”对目标问题的解进行编码，并尝试通过对这种“基因形式”的“交叉组合（crossover）”或“变异（mutation）”来形成对解空间的高质量搜索，从而快速地得到比较好的解。

在原始的遗传算法中，“基因表示”通常是一个二进制串。对于某些特定的问题，基于二进制串的“交叉”和“变异”会带来一些不必要的问题，如二进制串表示导致的具体信息的丢失，变异后的二进制串无法对应问题域的可行解等。因此，对于不同的问题，遗传算法通常会针对问题而引入一些针对“交叉”和“变异”规则以及基因表示的调整。

2 遗传算法

遗传算法作为一种优化算法，主要由两个部分组成：选择（Selection）和遗传（Genetic Operation）。选择步骤中，通过对现有解的评估，选取比较好的解作为生成下一组解的依据，即父辈代际（Parent）；遗传步骤中，根据选择得到的父辈代际，通过基因的交叉组合和突变等遗传算子（Genetic Operator）的操作，生成子辈代际（Children），作为下一个迭代的输入。

针对一个新的问题，从遗传算法的角度切入，我们通常需要设计的内容有：

1. 这个问题的解的基因表示的形式（Genetic Representation）
2. 这个问题的解的评价函数（Fitness Function）

3. 针对该问题的基因交叉组合的方式
4. 针对该问题的基因变异的方式

其中，交叉组合和变异是以基因表示形式为基础的。在遗传算法中，基因的交叉组合的方式可以认作是一个算法的核心步骤，它的作用是以两个比较好的问题解作为参照，以一定的规则保留这两个问题解的可能具备的好的特性，从而获得新的高质量的问题解。基因变异则起到引入随机噪声的作用，利于在已有问题解的邻域搜寻可能存在的局部更优解。

综上所属，遗传算法的基本框架如下：

1. 初始化：以一定的代际种群大小随机地初始化一组可行解（基因表示）。
2. 选择进化：
    i. 对现有代际种群进行评估和选择
    ii. 取经过选择后的个体，进行基因交叉组合和突变，获得下一个代际种群
3. 终止条件：算法在重复步骤2一定次数或者达到某些评估要求后终止

在步骤2.1中，选取用于生成下一代际的父辈代际有多种策略：

1. 轮盘随机选择（Roulette Wheel Selection）：对代际种群按评估函数值进行排序，评估函数值的大小与被选择作为父辈代际的概率成正比
2. 精英选择（Elitism）：按一定比例选取评估函数值最高的个体，作为父辈代际的同时，直接进入子辈代际

尽管遗传算法在各个领域问题中有广泛的应用，其通过“自然选择”启发式地产生新的可行解的方式并不可靠，同时也被广为诟病。此外，问题规模的增大会导致算法中基因表示、交叉组合等步骤成本的线性增加，复杂的问题评估也会给算法带来比较大的计算成本。

3 遗传算法在TSP中的应用

3.1 TSP（旅行商问题）

旅行商问题是一个经典的图搜索问题：给定N个城市和它们之间的距离，求最短的通过所有城市的路径。

尽管有着非常简单的问题描述和问题形式，旅行商问题却是NP-hard的。简单的暴力搜索算法甚至达到了O(N!) 的计算复杂度。求该问题的精确解随着问题规模的增长通常会变得不可行。

3.2 TSP遗传算法

基因表示

针对TSP问题的基因表示有多种选择，最符合直觉的一种为路径表示（Path Representation），如假设有7个城市，将城市按1-7编号，则城市之间的一条路径即作为该问题的一个解的基因表示（1—5—4—3—7—2—6）。

交叉组合

随机的交叉组合会导致生成的解不是可行解的情况发生（如路径重复地经过某个节点，或错过某些节点）。因此我们需要针对TSP制定一些交叉组合的规则，此处以“部分映射交叉”（Partially Mapped Crossover）为例：

$$P_1 = 1\ 2 \mid 3\ 4\ 5 \mid 6\ 7$$

$$P_2 = 7\ 2 \mid 6\ 3\ 5 \mid 4\ 1$$

将P_1的部分基因按位置拷贝到子个体中：

$$O = X\ X \mid 3\ 4\ 5 \mid X\ X$$

剩余的部分，按P_2进行填充（留空重复的基因）

$$O = 7\ 2 \mid 3\ 4\ 5 \mid X\ 1$$

以P_2为模版，逐个寻找填充剩余的部分（从剪切的片段右端开始，递增数字进行尝试直到可行位置，即4—5—6逐个尝试，直到找到6这个可行的数字为止）。

$$O = 7\ 2 \mid 3\ 4\ 5 \mid 6\ 1$$

变异

与交叉组合类似，随机的变异也会导致不可行解的产生。针对TSP的一种常用的变异方式为随机选择基因序列中的两个数字交换，即可得到新的可行解。

评估函数

显然地，为解决TSP问题我们希望找到路径长度尽量短的可行解，我们可以以路径长度的倒数作为评估函数。这个选择有两个优点：评估函数的值在0-1之间；在以评估函数的值作为概率选择父辈代际时，长度短的路径占的比例相对要更大一些，使选择步骤更倾向于选择更优的解。

3.3 变种

针对TSP的算法变种很多，此处列举较经典的几种。

环交叉（Cycle Crossover）

环交叉尝试找到两个父个体中相同的子城市序列，并按其中一个父个体的位置拷贝到子个体中，剩余的位置则按照另一个父个体的城市序列填充：

P_1= 1 **3** 5 **6** **4** 2 8 7

P_2= 1 4 2 3 6 5 7 8

O = 1 **3** 2 **6** **4** 5 7 8

部分映射交叉和环交叉都是保留城市的绝对位置的交叉算子，下面是一些保留城市间相对位置的交叉算子。

有序交叉（Order Crossover）

与部分映射交叉类似，有序交叉从父个体选取一段子串直接拷贝到自个体中，与部分映射交叉不同的是，有序交叉按城市在另一个父个体中出现的顺序填充剩余的部分：

P_1= 1 2 | 3 4 5 | 6 7
P_2= 7 2 | 6 3 5 | 4 1
O = 2 6 | 3 4 5 | 1 7

按序交叉（Order-based Crossover)

与环交叉类似，按序交叉首先找到两个父个体中的相同子序列，不同的是，通过直接改变其中一个父个体中这个子序列中元素的绝对位置得到自个体：

$$P_1 = 1\ 2\ \underline{5}\ 6\ \underline{4}\ \underline{3}\ 8\ 7$$

$$P_2 = 1\ \underline{4}\ 2\ \underline{3}\ 6\ \underline{5}\ 7\ 8$$

$$O = 1\ \underline{5}\ 2\ \underline{4}\ 6\ \underline{3}\ 7\ 8$$

除了按交叉方式变种外，还有不同的基因表示方式，如基数表示（Ordinal Representation）和邻接表示（Adjancency Representation），此处不再赘述。

4 实验结果与分析

我们以30个城市的环形地图为例，分别以CX、PMX、OX等算法进行实验并对结果进行分析。

实验均采用Elitism选择策略，0.02的变异率，0.4的精英率，种群大小为50。

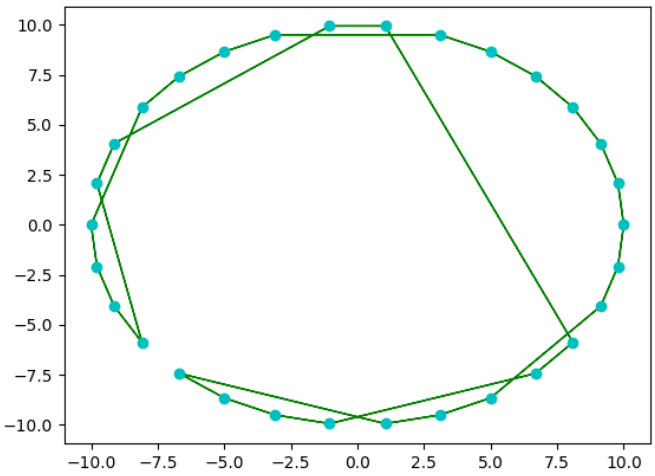


Fig1. Partially Mapped Crossover

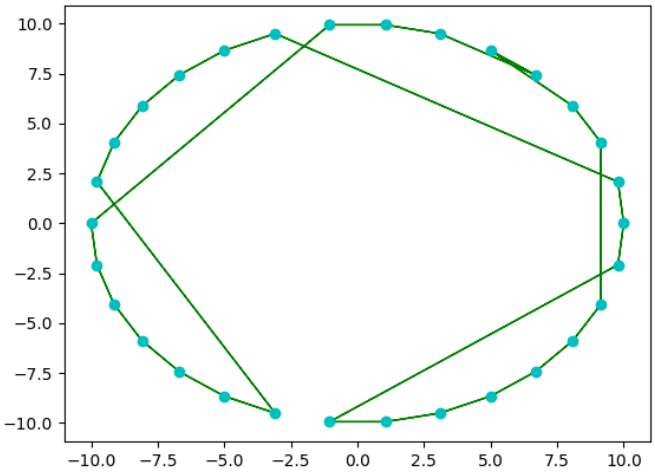


Fig2. Cycle Crossover

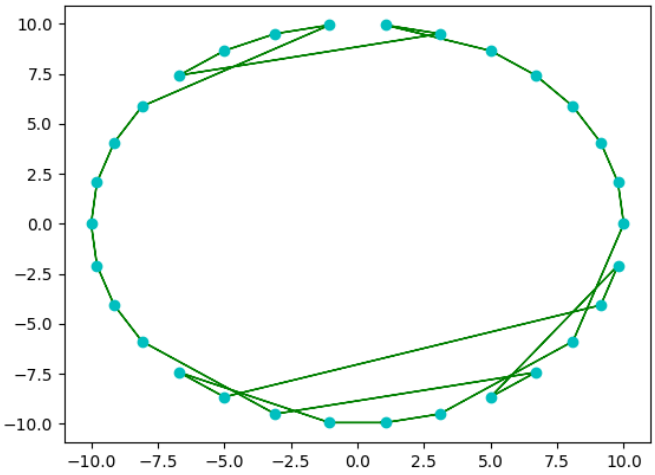
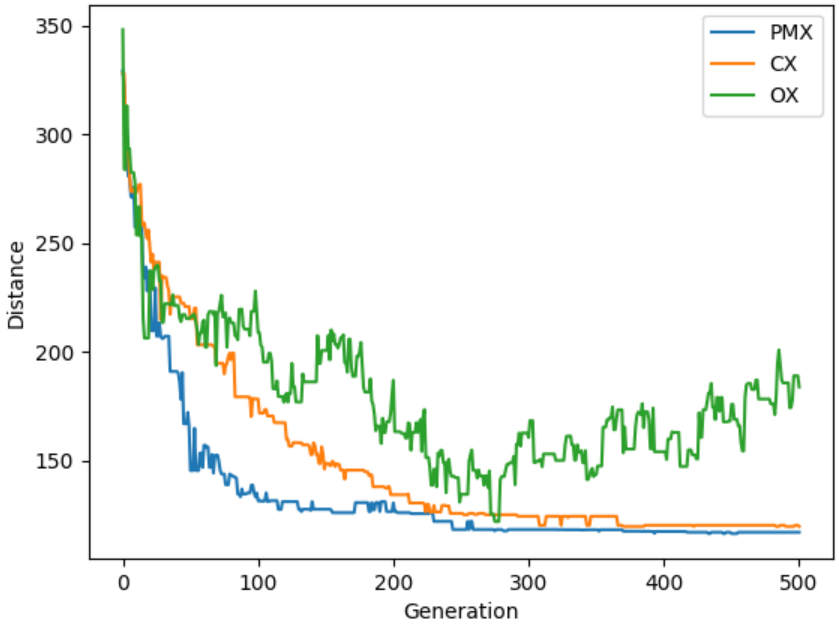


Fig3. Order Crossover



可以看到PMX和CX都很好地收敛了，PMX的收敛速度要更快一些，OX则更不稳定，但最优情况也达到了良好的结果。

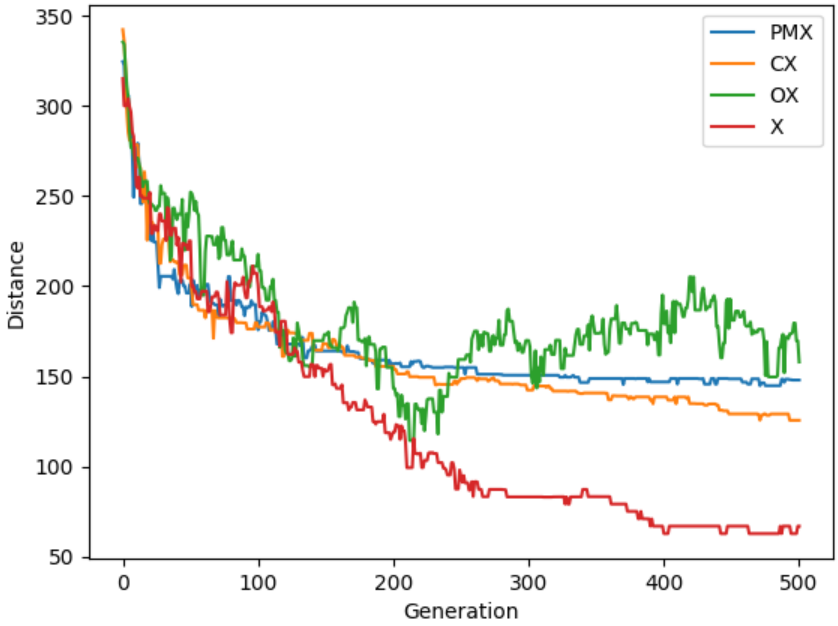
算法改良

我们在模仿PMX剪切基因后，同时尝试保持剩余城市在另一父个体中的相对位置和绝对位置，可以在该示例问题中达到更好的效果：

P_1= 1 2 | 3 4 5 | 6 7

P_2= 7 2 | 6 3 5 | 4 1

O = 7 2 | 3 4 5 | 6 1



5 总结

遗传算法在TSP领域的应用和研究在过去数十年内并未中断，与遗传算法在这一问题上表现出的出色的鲁棒性不无联系。除了本文提到的几个交叉算子外，一些新颖而复杂的交叉算子和基因表示仍然在不断被设计和提出。遗传算法在行程规划问题领域表现出的良好特性仍待分析和解释。实际上，正是“基因表示”和“交叉”、“变异”的晦涩性令一些研究者更倾向于使用有良好的理论支撑的如模拟退火等其它启发式算法。从这个角度，遗传算法的采用的原理的一般化是亟待攻克的。

参考文献

[1] Wikipedia contributors. "Genetic algorithm." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 18 Apr. 2019. Web. 1 May. 2019.

[2] Potvin, Jean-Yves. "Genetic algorithms for the traveling salesman problem. " Annals of Operations Research. 63. 339-370. 1996.

[3] Wikipedia contributors. "Travelling salesman problem." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 1 May. 2019. Web. 1 May. 2019.