湖南大學

HUNAN UNIVERSITY

工程优化技术课程作业

基于遗传算法的多旅行商问题求解

姓　名：苏凡珈

学　号：S2302W0250

班　级：2311

2024 年 8月

基于遗传算法的多旅行商问题求解

一、旅行商问题的描述

旅行商问题(Traveling Salesman Problem，TSP)是一个经典的组合优化问题。经典的 TSP 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。

多旅行商问题(MTSP)则是 m 个推销员分别访问部分城市，使除起点以外的每个城市仅被一个推销员访问一次，最终求遍历完所有城市的最低消费(距离、时间等)。当 $m=1$ 时，MTSP 就转化为经典 TSP，因此 TSP 是 MTSP 的一种特殊情况。现在像物流“最后一公里”等很多实际问题会涉及到多项任务的分配和优化，所以越来越多的问题被建模为 MTSP，因此 MTSP 受到越来越多的关注和研究。

给定一个简单的 TSP 问题，我们起初会采用枚举法排列出全部可以选择线路，然后运算出每条线路的长度，根据要求计算总路程的长度，最后比较走不同路线所产生总路线的长短，并从中找出最短的一条完整路径。排列路径流程其实就是对城市顺序进行排列然后组合的过程。不难看出，当城市的数目比较小时，很容易用枚举法就得到结果，但是随着城市数目规模的增加到 n，可枚举的路线表示为 $(n-1)!/2n$ 条，计算量还包括每一条路径的长度以及求出 n 个距离之和，当 n 比较大时，就会超出计算机无法承受的地步，出现“指数爆炸”的现象。因此利用枚举法求解城市数目较多的 TSP 问题将没有任何实际意义。而事实上，各种最优算法的计算时间随着问题规模的增大会以指数速度而增加，也远远超过计算机的计算能力，所以人们开始转向寻求有效的启发式算法。这是因为启发式算法使得问题在可接受的时间和空间复杂度的限制中去寻找最优的解，但是无法保证求得解的有效性和最优性。目前 TSP 问题的求解方法主要以禁忌算法、模拟退火算法、遗传算法、蚁群算法等为代表的现代智能优化算法。这些算法虽然思想各异，但是存在一个共同的目标：求得诸如旅行商问题等组合优化问题中的难解类问题的全局最优解。

二、旅行商问题数学模型

TSP 问题数学模型定义如下：

$$\min Z=\sum_{i=1}^{n}\sum_{j=1}^{n}r_{ij}x_{ij} \tag{1}$$

$$s.t.\sum_{j=1}^{n}x_{ij}=1,(i=1,2,...,n) \tag{2}$$

$$\sum_{i=1}^{n}x_{ij}=1,(j=1,2,...,n) \tag{3}$$

$$\sum_{i,j\in S}x_{ij}\le P-1,2\le|P|\le n-2,P\subset\{1,2,...,n\} \tag{4}$$

$$x_{ij}\in\{0.1\};(i,j=1,2,...,n) \tag{5}$$

公式（1）代表目标函数，r_{ij}代表城市i到城市j的权值；公式（2）、（3）代表回路中的任意一个节点均都会被访问且仅会被访问一次；公式（4）代表旅行商在所有节点的集合中不构成回路，P代表集合P的元素个数；公式（5）代表在旅行商访问路径中，若城市i到城市j未完全遍历，则变量$x_{ij}=0$；若城市i到城市j已被遍历，则变量$x_{ij}=1$。

MTSP 问题数学模型定义如下：

$$\min Z=\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}x_{ij} \tag{6}$$

$$s.t.\sum_{j=1}^{n}x_{ij}=1,(i=1,2,...,n) \tag{7}$$

$$\sum_{i=1}^{n}x_{ij}=1,(j=1,2,...,n) \tag{8}$$

$$u_i-u_j+nx_{ij}\le n-1(2\le i\ne j\le n) \tag{9}$$

$$\min Z=\sum_{i=1}^{n+k-1}\sum_{j=1}^{n+k-1}d_{ij}x_{ij} \tag{10}$$

$$s.t.\sum_{j=1}^{n+k-1}x_{ij}=1,(i=1,2,...,n) \tag{11}$$

$$\sum_{i=1}^{n+k-1} x_{ij} = 1, (j = 1, 2, ..., n) \quad (12)$$

$$u_i - u_j + nx_{ij} \le n-1 (2 \le i \ne j \le n), x_{ij} = 0, i, j \in \{1, n+1, ..., n+k-1\} \quad (13)$$

公式（6）表示路径最小的目标；公式（7）表示访问城市 j 之前需有一个已访问过的城市；公式（8）表示访问城市 i 之后需有一个即将访问的城市；公式（9）中 u_i 为访问城市 i 次序的取值范围 $\{0,1,...,n-1\}$；在公式（10）~（13）中：d_{ij} 为城市 i 到城市 j 的距离，x_{ij} 为 0 或 1，其中 1 表示从城市 i 到城市 j 的方向，0 表示从城市 j 到城市 i 的方向；把首个城市当作起点城市，如果有 k 个旅行商，则可将多旅行商问题转换为单旅行商问题，即添加 $k-1$ 个与首个城市相同的城市；用图论的观点解释多旅行商问题本质上即为在图 G 中找寻一个权值最小的汉密尔顿回路。

三、遗传算法

遗传算法（Genetic Algorithm, GA）的起源理论基于进化论与遗传学说。

进化论中的自然选择学说表示物种的产生是自然选择的必然结果。适应环境能力较强的个体才能存活并去繁衍后代，而适应环境能力较弱的个体则将被淘汰。

遗传学说的基因遗传原理认为遗传以基因的方式置于染色体内。交叉与突变能够产生更适应于环境的后代个体。经过优胜劣汰的选择过程，适应性较高的基因型得到保留，适应性较低的基因型则被淘汰。

借鉴自然选择学说与基因遗传原理，遗传算法将生物界进化的过程运用于科学研究。作为一种具有全局搜索方式的算法，遗传算法借鉴生物界适者生存与染色体随机繁衍规则，给解向量进行编码后形成初始种群，再进行选择、交叉、变异等操作过程，从而得到种群的优化解。

1.遗传算法基本要素

（1）个体编码（Parameter Coding）

遗传算法采用编码和解码的运算方式来表示目标问题，使得目标问题具有染色体基因型的结构。编码方式决定遗传操作方式，路径表示法是常见的编码方法，在遗传算法中已经证明了其收敛性。在求解 TSP 及 MTSP 问题中，编码方法也

常采用路径表示法。

（2）初始种群（Initialing Population）

在遗传算法初始，由若干个个体组成的群体即为初始种群。由初始种群出发，遗传算法进行遗传操作和进化。初始种群可由一定的随机方法来生成。

初始种群为遗传进化的第一代父代，遗传算法的运行过程与初始种群的规模大小有一定的联系。如果种群规模过大，则会增加计算量，从而影响算法性能，并会对交叉操作过程产生负面作用，很大程度上会影响基因配对的多样性；但如果种群规模过小，则可能会导致遗传算法的早熟现象发生，即算法的搜索过程会一直停留在未成熟的阶段。

（3）适应度函数（Evaluation Function）

适应度是物种对环境适应程度的一个衡量标准，对环境适应度较高的物种会被选择繁衍后代，而对环境适应程度较低的物种则繁殖后代的机会较少，甚至会灭绝。度量每个个体适应度值的函数即为适应度函数，在遗传算法中一般用目标函数的倒数作为适应度函数，适应度值的高低直接影响个体基因被遗传到子代的几率。

（4）遗传操作（Operators）

遗传操作是遗传算法最核心的部分，是对种群中个体按一定规则进行操作后，实现择优的进化过程。通过选择操作，用高适应度的个体替换低适应度的个体，保证搜索朝着优化的方向进行；通过交叉操作，将父代个体以概率 P_C 进行基因交叉互换后产生子代个体，保证种群的多样化；通过变异操作，将父代个体以变异概率 P_m 进行基因变异后产生子代个体，从而加速遗传过程向最优解收敛。遗传算法的选择、交叉、变异搜索过程，使问题的解一代代地进化，最终得到趋近于最优解的次优解。

（5）选择算子（Selection）

选择算子从种群中挑选适应度较强的个体进行下一步的交叉、变异操作。其中，轮盘赌选择法（Roulette Wheel Selection, RWS）是遗传算法中最早提出的一种方法。该策略将种群中的每一个个体作为圆盘当中的一块，转动圆盘，当圆盘停止转动时，指针所指的个体则会被选中。显然适应度值越高的个体被选中的几率越大。设种群大小为 M，种群中的某个个体 i 的适应度值为 f_i，则该个体 i 被选

择的概率 P_i 为：

$$P_i = \frac{f_i}{\sum_{i=1}^{M} f_i} (i = 1, 2, ..., M)$$

（6）交叉算子（Crossover）

交叉算子决定遗传算法的全局搜索能力。交叉算子借鉴生物有性繁殖中的基因重组过程，将父代中有优势的基因进行交叉互换后遗传给子代个体。传统遗传算法的交叉算子有 PMX、OX、CX 等。

（7）变异算子（Mutation）

变异算子决定遗传算法的局部搜索能力。变异算子借鉴生物细胞分裂的基因突变过程，以较小的概率对个体编码串的基因位进行细微改变，其目的是较好地维持种群的多样性。传统遗传算法的变异算子有 Swap、2-opt 等。

2.遗传算法流程

遗传算法将选择、交叉、变异等过程作用于种群之中，并迭代地重复进行操作，程序最终会得到问题的满意解。用遗传算法求解一个实际问题的步骤如下：

Step1：确定编码策略与遗传参数；

Step2：确定适应度函数的表达方式；

Step3：根据问题模型设计遗传算子；

Step4：用特定的构造方法初始化种群；

Step5：计算种群中个体的适应度函数；

Step6：用选择操作挑选若干个体作为父代；

Step7：对父代个体的基因段进行交叉操作；

Step8：对父代个体的某些基因进行变异操作；

Step9：进化到一定代数后记录最优个体；

Step10：种群更新，重复 *Step5~Step9*；

Step11：进化终止，得到次优解。

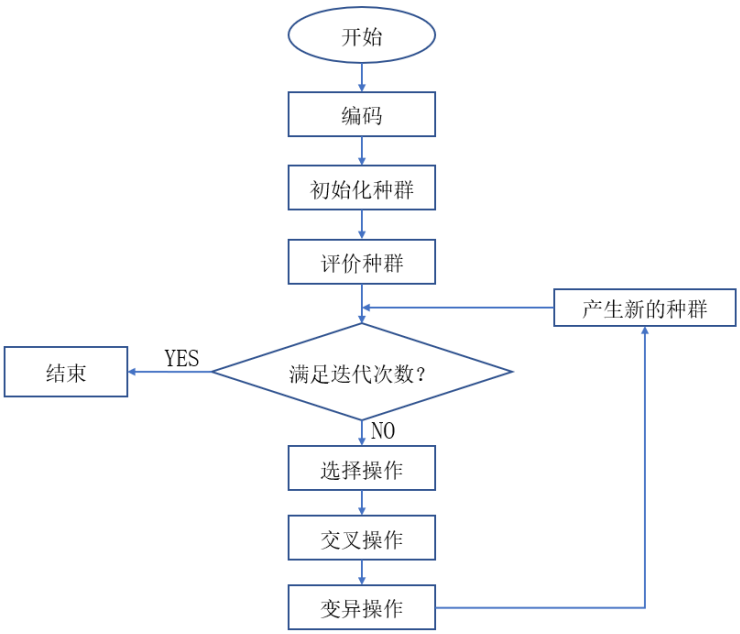

图 1 遗传算法流程图

四、算例及结果分析

1.TSP 算例

首先产生 50 个随机样本点代表 50 个城市，如下图分布。

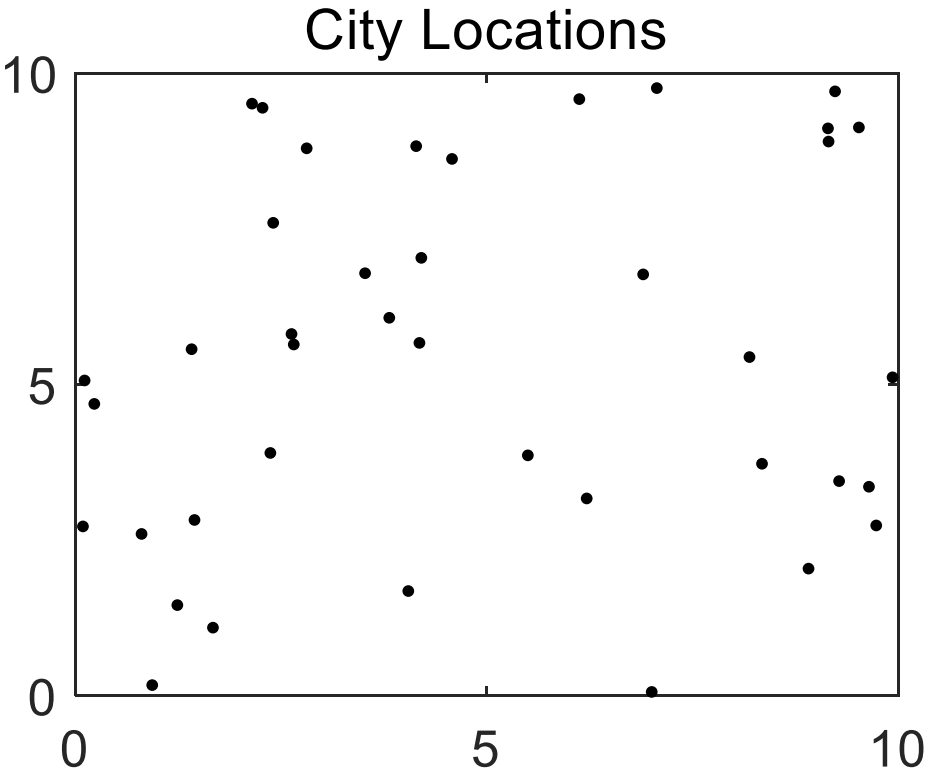

图 2 城市分布

（1）编码方式。我们这里使用十进制编码方式。把旅行商途经的城市的顺序序号作为遗传算法的编码。假设 50 个城市的序号为$\{1,2,...,50\}$，那么它的任意一个全排列就是一个数据编码，表示一个染色体，每个染色体就代表一个解，不同解是靠染色体上不同的基因决定的。

（2）适应度函数。由于旅行商问题的规划模型的目标函数是要求解最小值，所以适应度函数就用路径的总长度的倒数来表示。这样，符合了遗传算法的优胜劣汰的搜索策略。

（3）初始群体的产生。随机生产一个大小为$N=100$且每条染色体上的基因长度都是的初始种群，作为第一代个体。

（4）选择过程。我们需要构造一个合适的隶属函数。然后计算出种群中每个遗传个体的适应度，从而得到种群中遗传个体适应度的和。那么我们就可以用来表示个体被选中的概率。

（5）交叉过程。首先设定交叉概率$P_c=0.9$。其次，根据这一概率，选出要进行交叉操作的个体，并将它们两两配对。然后在染色体上$1\sim n-1$个基因编号之间随机地选出两个，之间的基因作为接下来将要进行交叉的对象。

（6）变异过程。进行完交叉操作后，以变异概率$P_m=0.2$。从群体中选出要进行变异的遗传个体。然后，随机地从$1\sim n-1$之间选泽两个数作为变异点，将基因进行交换。

实验结果

使用 MATLAB 进行编程验证，得到结果如下：

1）迭代次数为 100 时，得到最优路径如图 3，此时最短距离为 73.028。

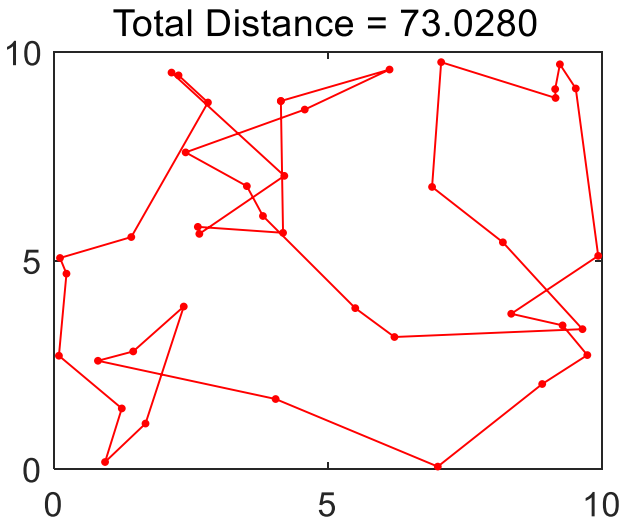

图 3 迭代次数为 100 时最优路径

2）迭代次数为 500 时，得到最优路径如图 4，此时最短距离为 55.9223。

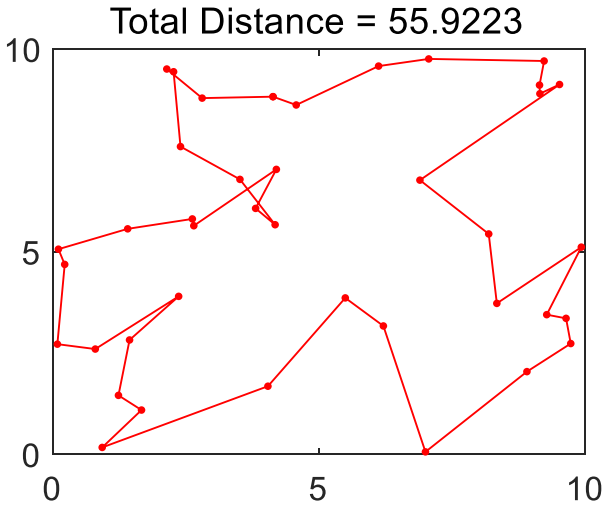

图 4 迭代次数为 500 时最优路径

2）迭代次数为 1000 时，得到最优路径如图 5，此时最短距离为 54.8064。

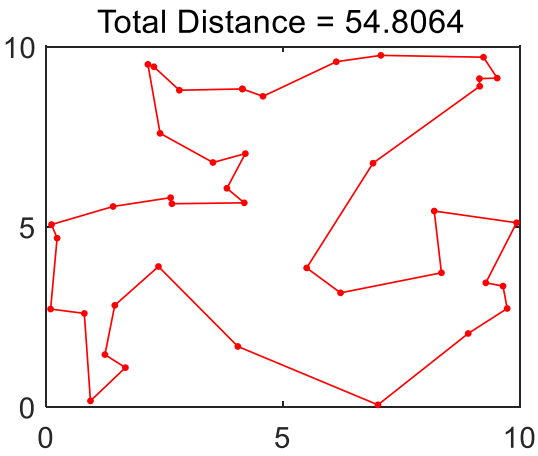

图 5 迭代次数为 1000 时最优路径

2.MTSP 算例

本算例使用样本的城市坐标与上述 TSP 算例相同，城市总数为 50，假设有 5 个旅行商，每个旅行商至少要经过 8 个城市。

在交叉操作时，首先生成 4 个断点，将父代染色体分割成为 5 个旅行商行走的路径。产生两个随机数 $r_1, r_2 \in [0,5)$，如 $r_1 = 0, r_2 = 1$，则将旅行商 1 与旅行商 2 进行交叉；计算旅行商 1 与旅行商 2 中旅行商访问城市少的城市个数，如旅行商 1 访问城市个数为 8，旅行商 2 访问城市个数为 10，则选择 L=8（L 为需交叉的路径长度）；产生两个随机数 $q_1, q_2 \in (1,8)$，如：$q_1 = 2, q_2 = 5$,则对旅行商 1、2 的第 2~5 个基因进行交叉互换。

实验结果

使用 MATLAB 进行编程验证，得到结果如下：

1）迭代次数为 100 时，得到最优路径如图 6，此时最短距离为 79.4837。

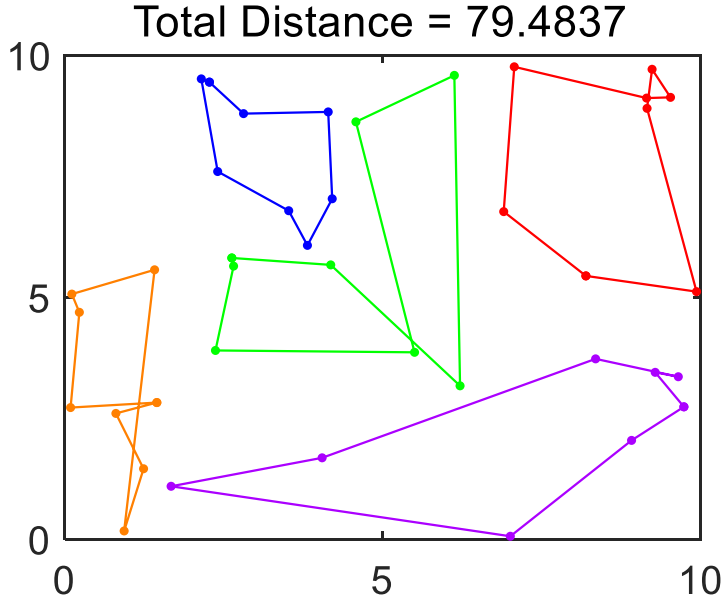

图 6 迭代次数为 100 时最优路径

2）迭代次数为 300 时，得到最优路径如图 4，此时最短距离为 69.2977。

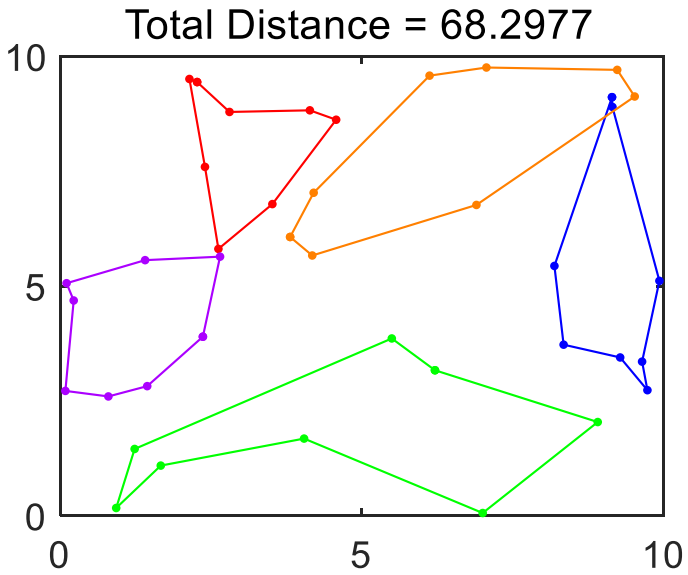

图 7 迭代次数为 300 时最优路径

2）迭代次数为 500 时，得到最优路径如图 5，此时最短距离为 60.8777。

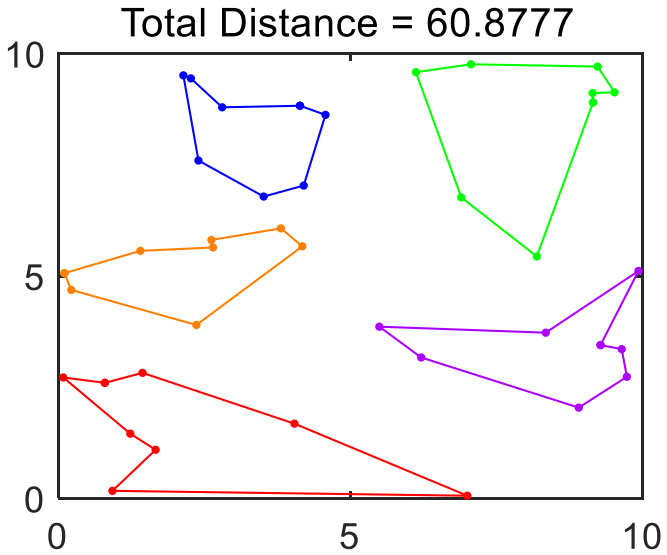

图 8 迭代次数为 500 时最优路径

从程序运行结果可以看出，由于遗传算法的启发式特性，每次运行的结果大都不相同，但都相近，在进化到 500 代左右时，总距离基本已经可以达到最小值，再进化更多代数总距离无明显改进。可见遗传算法对于求解 TSP 问题以及 MTSP 问题具有良好的效果。