**遗传算法经典解决--旅行商问题**

旅行商问题：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。选择行进路线，以使总的行程最短。

1. **问题建模**

旅行商问题可以转化成求一条与起始点无关的最短回路路径的问题。假设个城市，矩阵是一个的方阵，表示i城市到j城市的距离，问题便可以先对个城市随机标号，再求比较的值选择不同标号下的最小值。

1. **遗传算法中对应的步骤**

初始化种群：设定种群规模，对应个体染色体用的数组（长度为个的数组）表示路线（下标临近的城市生成对应路径）。以随机生成二维数组初始种群。

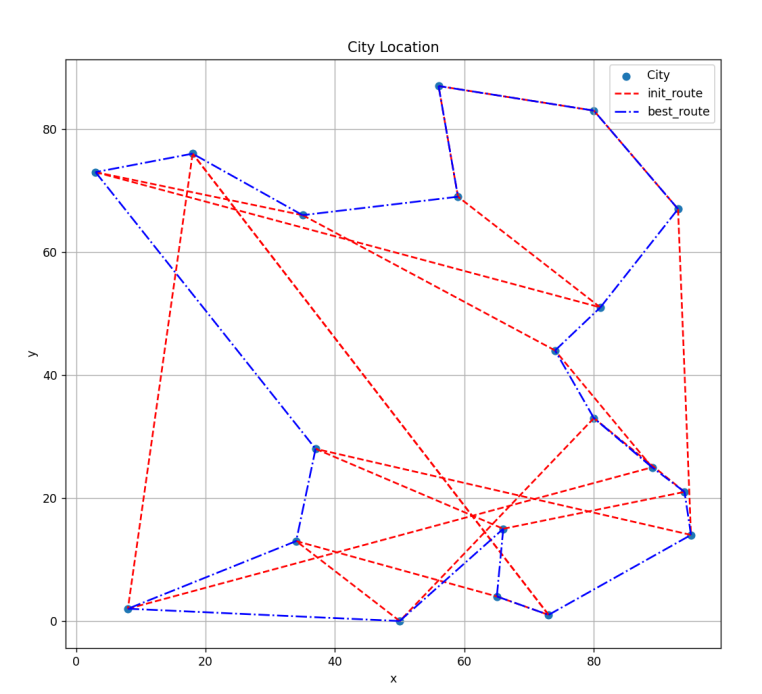
适应度计算：建立的映射关系，生成一维数组。

选择：选择的逻辑是优胜劣汰，不妨直接从适应度大的个体中筛选个体，因此对降序排序，筛选前 规模的种群作为选择对象。

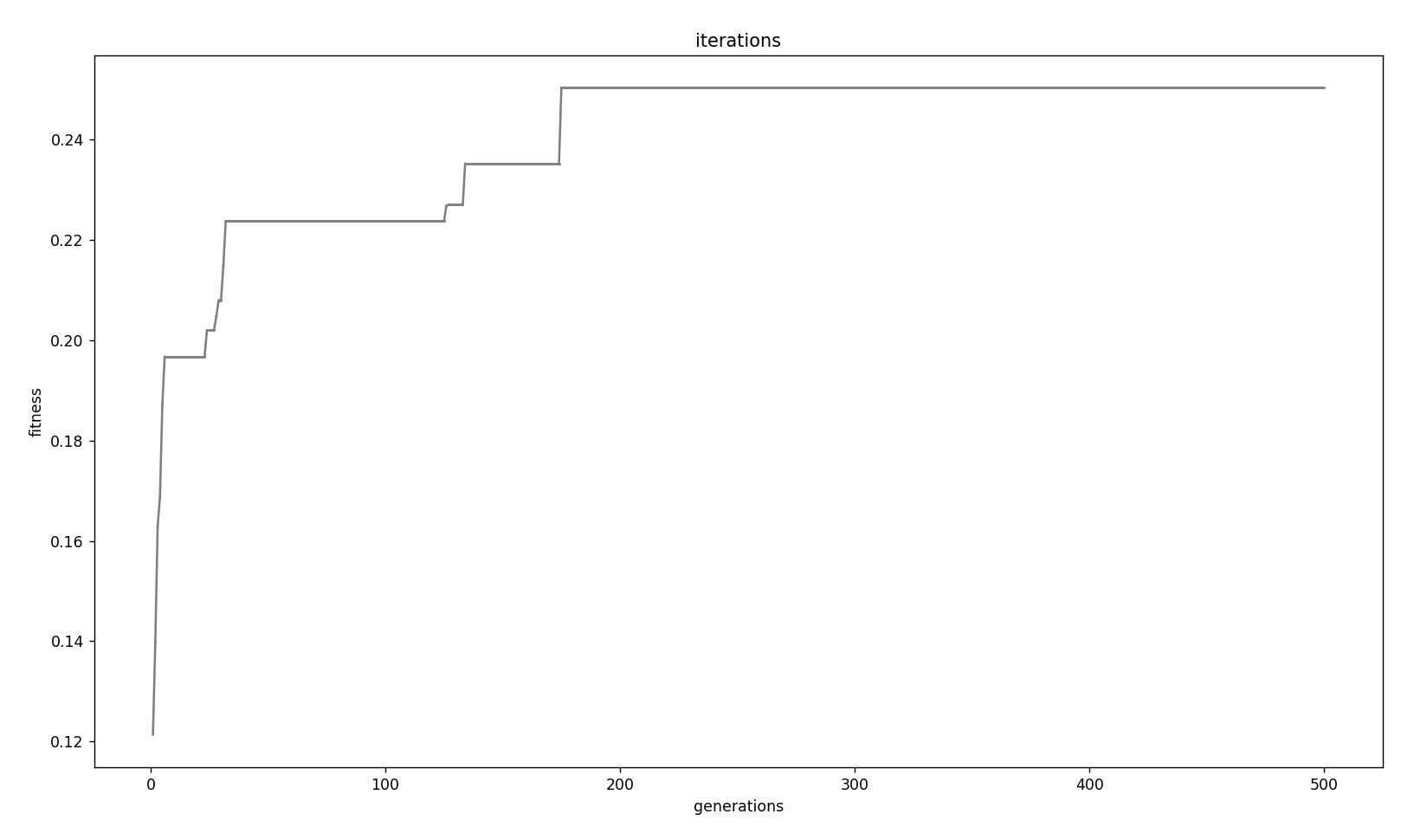
交叉：首先在中随机找两个父辈。由于旅行商问题的特殊性，染色体简单的片段交换，可能会使某个城市走多遍，不满足问题要求。并且由于在此处等价，所以此处采取先随机两个不相同的的数截取parent1从到的片段，再在中顺序取出前片段中不存在的基因信息前后连接生成个体。最后对上述过程重复次，生成对应数量的子个体。

变异：由于selected\_population在这一代中已经足够优秀，不妨我们仅对子个体变异处理。由于基因特殊性，不能单个基因突变，我们采取基因位置交换的方法，对每个基因满足概率是随机交换基因。进行该处理后我们便得到下一代种群，之后不断迭代即可。

1. **旅行商问题在python中的可视化结果**

如图是在上述条件下随机生成的城市坐标图，红色线表示的是初始第代的最优解，蓝色线表示的是迭代次后收敛的最优解。是横坐标为迭代次数，纵坐标为当前迭代次数下的最优适应度。可以发现当规模 情况下，适应度得到很好地收敛，从最开始的总长度 最后优化到总长度

，给出最优城市回路路线[3, 17, 11, 9, 19, 4, 6, 12, 1, 16, 15, 13, 2, 7, 8, 10, 18, 0, 14, 5] 。



**基于遗传算法实现日常任务顺序安排**

日常任务顺序安排问题导入：日常生活中我们对事情的安排往往依靠感觉，但是这种感觉是有依据的，基于我们明白难以在一定时间完成所有任务，所以我们需要排序实现收益最大化。

1. **问题建模**

上述问题可以转化成，时间内，有项任务，应该在个时间段分别做什么任务来达到收益最大。

但是这样问题仍然描述不够具体，缺少量化的指标，来建立和适应度的映射。通过周边人分享和个人经历判断，我们思考一个问题的先后顺序主要基于以下六点：重要性；紧要性；时间耗费程度；关联性；有趣性；困难性，通过下面六点的综合作用分析一个问题的适应度大小来排序问题的先后。比较六点对先后顺序的影响我们创建任务的数据类型包含上述六点：(存储与其他每一个任务的关联度,可以用范围的数值表示，数组大小)。

下面简单构建上述六点和适应度映射关系。都可以用范围的数值表示，表示完成该任务所需的时间，所以对应每一时刻做该任务相当于完成 的进度，因此同一任务每一时刻对应应该有个固定一个系数，然后再乘上（设其他任务在该时刻前的完成度），即算出当前时刻的适应度，则。

1. **遗传算法中对应的步骤**

初始化种群：设定种群规模，对应个体染色体用的数组（长度为个的数组）表示对应时间完成对应任务的下标。以随机生成二维数组初始种群。

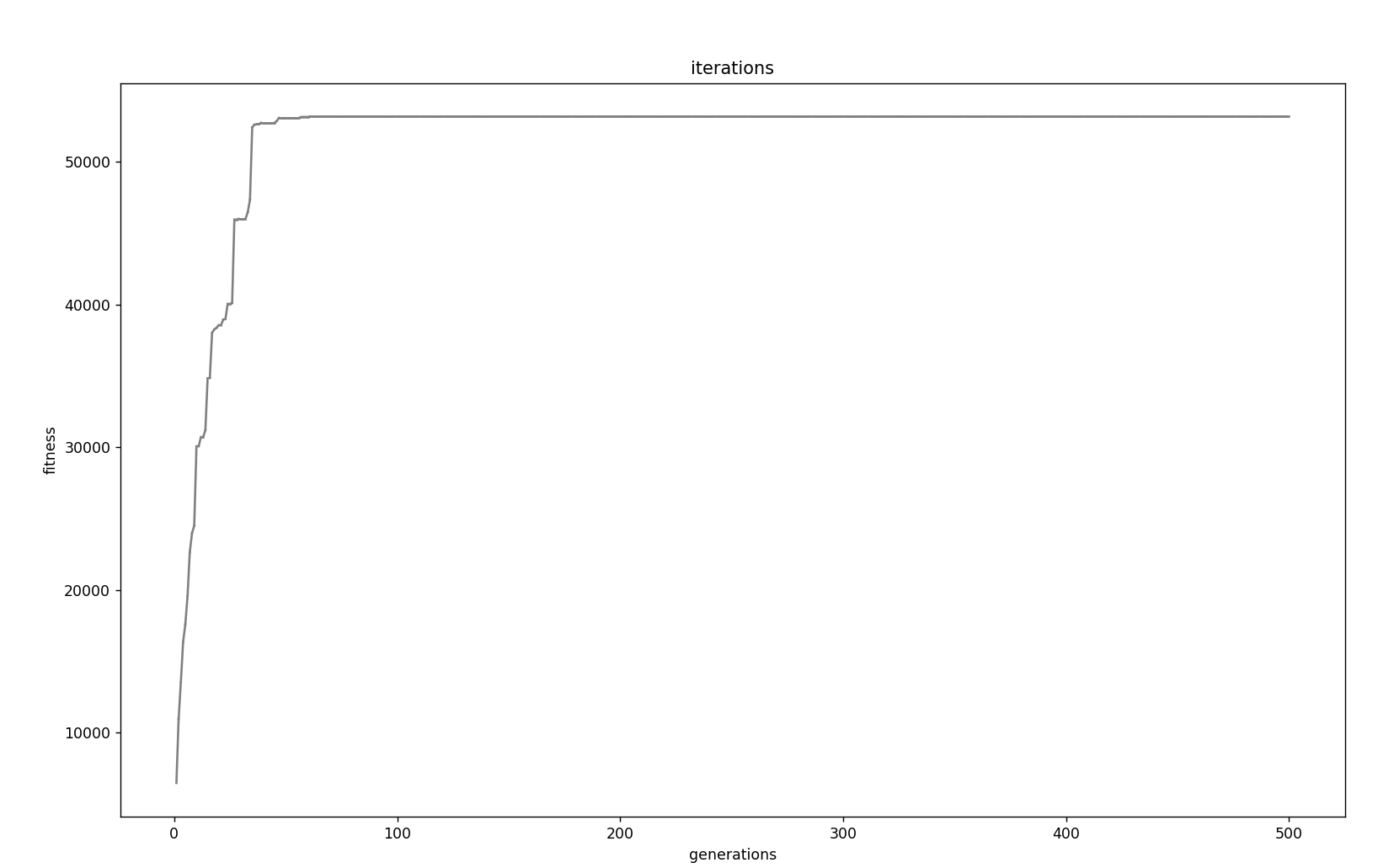
适应度计算：计算个体对应的适应度即可（再乘上一个对应的系数是因为直接算出来的适应度太小，这个系数要随规模的扩大指数增长）

选择：同旅行商问题，对降序排序，筛选前 规模的种群作为选择对象。

交叉：首先在中随机找两个父辈。然后两父辈进行基因片段交换，由于交换后可能出现做某个任务时间超出需要时间，所以需要加以判断，去除不符合的。

变异：同旅行商问题，保留selected\_population，对子代做旅行商问题相同处理，即可得到下一代种群，最后再不断迭代即可。

1. **日常任务顺序安排在python中的可视化**

如图，从初代适应度到最后收敛到筛选出最佳完成任务顺序（由于Task数组过于复杂，放在附录中）。