基于GA遗传算法求解市辖区内多起点防疫物资发放路径规划问题

摘要：全国防疫政策放开初期，感染人数剧增，新冠防治药物等防疫物资紧缺。天津市人民政府拟以区为单位，向辖区内各社区发放一批医疗物资。本文将该问题抽象为MTSP多旅行商问题，并建立数学模型；以5个定点作为起点，30个随机点作为途经点，模拟天津市南开区内物资发放情形；选用GA遗传算法，编写Python程序，对该模拟问题进行求解；顺利求解出路径规划结果，并对其进行分析与讨论；最后提出了未来进一步研究方向。

关键词：MTSP多旅行商问题、GA遗传算法、路径规划、组合优化、防疫物资发放

1. 问题描述

全国防疫政策放开初期，感染人数剧增，百姓在短时间内的集中购买，导致新冠防治药物等防疫物资紧缺。为妥善满足百姓需求，天津市人民政府拟以区为单位，向辖区内各社区发放一批防疫物资。现以天津市南开区为例，假设待发放防疫物资集中在各地段医院，为尽快将物资发放到各社区，避免二次倒手，南开区政府决定为每一所囤积物资的医院安排一辆运输车，将物资直接从医院发放至各社区，跳过街道层级。每一辆运输车发放完成后，需返回所属医院，听从统一调遣；而每辆运输车运力有限，仅能负责有限数量的社区；为保证发放效率，需确保每辆运输车仅途径各社区一次，往返医院一次，便可完成发放任务；且各地段医院并非均布在各社区附近，无法提前确定各医院所负责社区。

求解：如何规划路线，能在满足以上要求的情况下，使辖区内各运输车走过的总路程最短？

2. 问题分析

2.1 问题抽象

根据问题描述，可将该问题抽象为MTSP多旅行商问题：以各医院作为起点，各社区作为途经点，各运输车作为多个旅行商；“发放完成后返回医院”，意味着各旅行商的路径均是闭合环路；“运力有限”，意味着各环路所包含的点不能过多；“仅途径各社区一次，往返医院一次”，意味着经过各点有且仅有一个环路；“无法提前确定所负责社区”，意味着途径点的划分与排序要同时考虑。

2.2 MTSP 问题简述

“MTSP 多旅行商问题”是一类经典的组合优化问题，类属运筹学范畴。它是传统“TSP 旅行商问题”的升级版，是“VRP 车辆路径规划问题”的简化版或松弛版。与传统 TSP 问题相比，MTSP 问题包含多个旅行商，须同时考虑途径点的划分与排序问题，在同等问题规模的情况下，MTSP 问题的复杂度较 TSP 问题明显提高，求解思路也不尽相同；与 VRP 问题相比，由于不精确考虑容量、时间窗等问题，MTSP 问题虽在复杂度方面与 VRP 问题相差不大，但在建模与求解方面，难度则有所降低。

近年来，TSP 问题及 VRP 问题的求解方法渐趋成熟。相比之下，关于 MTSP 问题的研究则略显单薄，成熟的参考资料也并不丰富。故本文选取 MTSP 问题，这一经典但又具有一定难度的运筹优化问题，作为研究对象；利用本学期所学及课外知识，力求为 MTSP 问题的研究尽自己的一点绵薄之力。

3. 数学建模

3.1 模型抽象

该问题可等效抽象简化为以下数学模型：

①将南开区辖区等效为→1000*1000 的二维平面

②将辖区内各地段医院等效为→平面 5 个固定点，作为各环路起、终点。该 5 点坐标依次为(250,750)，(750,750)，(500,500)，(250,250)，(750,250)

③将辖区内各社区等效为→平面内 30 个随机点，作为途经点。其坐标随机指定，由此即可保证 5 个起点与途经点分布的相对不均匀性，符合原题设要求。

由此，原问题即可等效抽象简化为→该 35 个点集范围内的 MTSP 问题。其中旅行商个数=5，即需规划 5 条闭合环路，每条环路以 5 个固定点其中之一作为起、终点（相对的概念），并使得 35 个点集中的各点属于且仅属于 1 条环路。在此前提下，求解使得 5 条环路总路程最短的路径规划。

易知，该模拟问题与原问题相比，虽问题规模及各点位置不尽相同，但问题原理上是完全等效的。故求解该模拟问题的算法也同样适用于求解原问题，本文后续便借助求解该模拟问题，来等效求解原问题。

需说明一点：30 个途经点的“随机性”是指其位置的生成机理是随机的，而非人为指定的。但一旦生成，30 个途径点的坐标便保持固定不动，与 5 个起点的坐标一并保存，作为本问题的数据集，后续的求解均是围绕该数据集展开的。不过求解过程中使用的算法是适用所有类同的数据集的，本文选定数据集只是为了便于论证算法正确性。

35 个数据点的位置如“图 1”所示，其中红色点表示 5 个起点，蓝色点表示 30 个途经点。

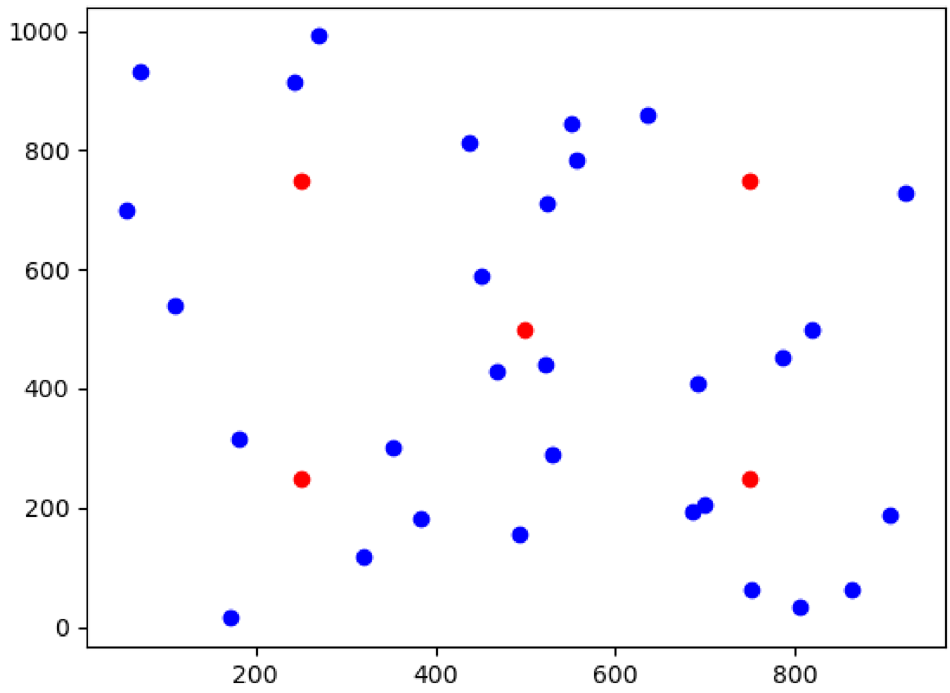


图 1

3.2 模型假设

现规定原问题满足如下假设：

1）各运输车容量相同，各社区物资需求量相同，且每辆运输车容量恰好=6*各社区物资需求量；故安排每辆运输车负责 6 个社区，恰好覆盖所有 30 个社区。

2）各点之间的路程用直线距离代替。

3）不考虑指定送达顺序要求、交通实时情况等无关因素。

基于以上假设，可将模拟问题进一步简化为：35 个点集内的 MTSP 问题，旅行商数=闭合环路数=5，每条环路仅包含 1 个固定起点，包含 6 个随机途经点。35 个点集中各点属于且仅属于 1 条环路。在此前提下，求解使得 5 条环路总路程最短的路径规划。

4. 算法选择

本文选用 GA 遗传算法求解该 MTSP 问题，GA 遗传算法类属启发式算法。现对该算法选择作解释说明。

4.1 为何选择启发式算法

MTSP 问题类属 NP 完全问题，复杂度极高，在问题规模较大时，应用传统精确式算法（如线性规划、动态规划、分支定界法），虽能确保得到全局最优解，但其收敛时间过慢，客观上无法实现；而启发式算法虽不能保证所得解一定是全局最优解，但可实现在多项式时间内快速收敛，适合大规模问题的求解。

原问题所求实际问题，背景选取为市辖区，其问题规模可想而知十分庞大；而大规模问题往往对解的最优性要求并非十分严苛，例如本问题，在市辖区范围内，总路程上几公里甚至几十公里的差距，均可忽略不计。模拟问题规模虽并非如原问题那般庞大，但为了寻找等效求解原问题的方法，也应考虑原问题的规模来选择算法求解模拟问题。故针对该模拟问题，应选用启发式算法进行求解，并且可以对算法进行优化，力求所得解尽量逼近全局最优解，弥补其不足。

4.2 为何从诸多启发式算法中选择 GA 遗传算法

针对路径规划问题，较为有效的主流启发式算法为蚁群算法和遗传算法。但蚁群算法由于其正反馈机制，并不十分适合多目标规划问题，常用于求解 TSP 问题或单起点的 VRP 问题。而本问题不仅为 MTSP 多旅行商问题，且为多起点的 MTSP 问题，规划目标过多，应用蚁群算法建模求解难度较大；而遗传算法并未对多目标问题显示出明显的不适应性，依然可以正常求解。故本文选取 GA 遗传算法来求解该多起点的 MTSP 问题。

5. 求解过程

5.1 GA 遗传算法流程

1） 定义个体基因型
2） 初始化构造种群

开始遗传迭代，预设迭代次数

3） 自然选择——定义目标函数，根据目标函数值计算个体适应度，适应度越高存活几率越大，刷新种群
4） 基因交叉（个体交配）——原种群个体两两交配，基因交叉重组，产生新的子代并替换父代，刷新种群
5） 基因变异——个体基因型发生随机变异，刷新种群

是否已达预设迭代次数？ No——返回步骤 3） Yes——进入步骤 6）

6） 遗传迭代结束，保存最终种群

7） 寻找最终种群中最优个体（适应度最高），计算其目标函数值，即为算法最终所得解

5.2 各步骤具体定义、推算

针对本文所求模拟问题，对遗传算法各步骤函数、变量进行具体定义

5.2.1 个体基因型 entity

个体基因型为 30 个途经点的排序。

如：entity_1=[10, 28, 15, 4, 6, 20, 23, 24, 29, 22, 12, 8, 26, 18, 21, 0, 27, 25, 13, 16, 7, 5, 2, 1, 3, 14, 19, 11, 17, 9]

5.2.2 目标函数 aimfunction 及适应度函数 fitness

目标函数为个体基因型所规划的 5 条环路路程之和——将个体基因数据 6 个为 1 组，分成 5 组；第 1 组对应第 1 个固定起点，与之构成第 1 条环路；第 2 组对应第 2 个固定起点，与之构成第 2 条环路；以此类推，规划出 5 条环路，计算总长度。

可知，本问题为目标函数最小化问题。目标值函数值越小，解的性质越好，对应个体的适应度应越高。故将个体适应度定义为个体目标函数值的倒数。

5.2.3 自然选择 selection

应用“轮盘赌法”进行自然选择，其思想为——个体被选中的概率与其适应度函数的大小成正比。用概率公式表示为：

$$p(x_i)=\frac{f(x_i)}{\sum_{i=1}^{n} f(x_i)}$$

其中，n 为种群中个体总数，f(x)为个体适应度函数值。

轮盘赌实现方法：生成 1 个[0,1]之间的随机数 r。若 $r \leqslant q_1$，则选择 x_1；若 $q_{i-1}<r \leqslant q_i$，则选择 q_i，其中 $2 \leqslant i \leqslant n$。$q_i$ 被称为染色体 i 的累计概率，其公式为：

$$q_i=\sum_{j=1}^{i} p(x_j)$$

5.2.4 基因交叉（个体交配）crossover

常规遗传算法常使用“单点交叉法”，通过下例易于理解：

父代基因型：

Father=10101111001010111O100

Mother=01000100110001111O101

单点交叉得到子代基因型：

Son=10101111000001111O101

Daughter=01000100111010111O100

而本文针对所求问题定义基因型的特殊性，由于基因型将划分成 5 段路径，为尽可能多地影响多段路径，故优化基因交叉方法为“随机双点交叉法”，其中 2 个交叉点位置随机确定，每个迭代轮次中的每个个体，交叉点位均不同，保证了最大化全局搜索，提高了最终解的全局最优性。通过下例易于理解：

父代基因型：

Father=10101111001010111O100

Mother=01000100110001111O101

单点交叉得到子代基因型：

Son=10101111110001111O100

Daughter=01000100001010111O101

5.2.5 基因变异 mutation

通过随机交换个体基因中 2 数据的位置来实现基因变异，通过下例易于理解：

Entity_old=10101111001010111O100

Entity_new=10100111001010111 1100

5.2.6 运行参数定义

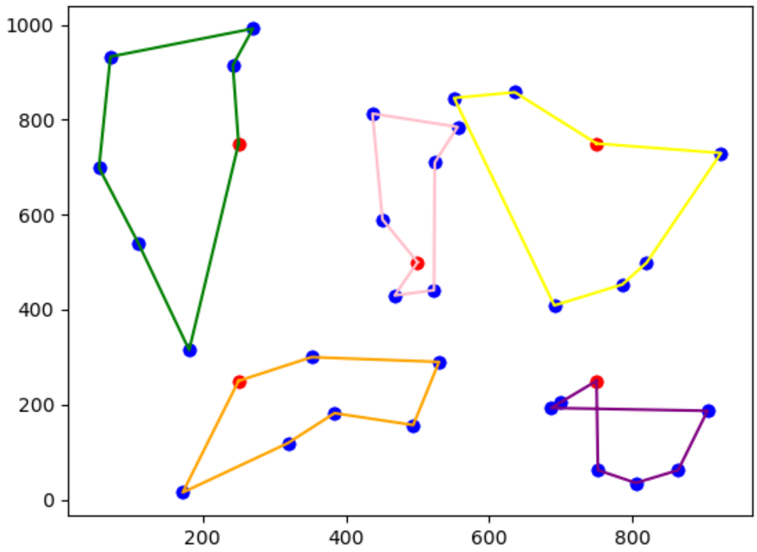
```
entity_num=100          #每个种群中个体数
GA_times=6000           #遗传迭代次数
cross_prob=0.4          #基因交叉概率
mut_prob=0.02           #基因变异概率
```

有关算法其他具体内容，详见附件中程序源码 MTSP_GA.py

6. 结果分析与讨论

6.1 算法所求得“最优解”及收敛过程

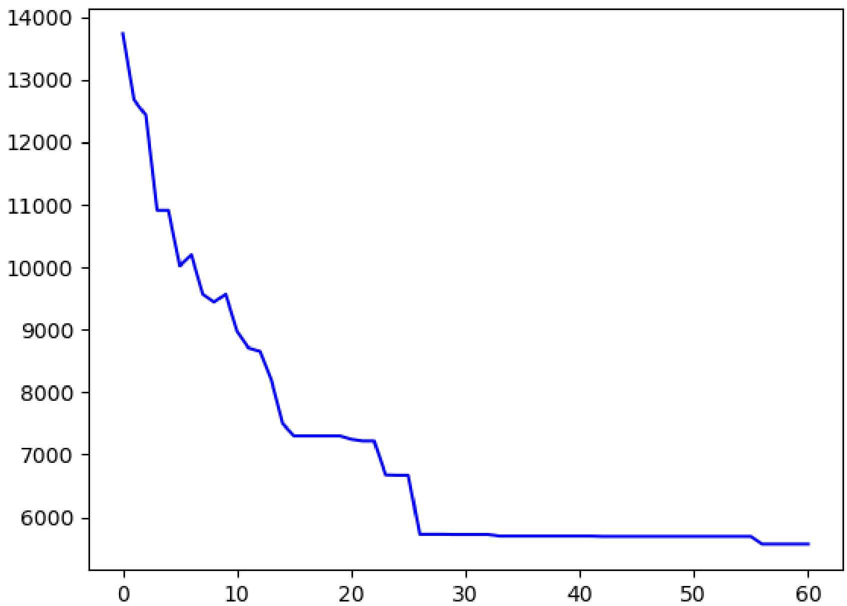
6.1.1 “最优解”



最优解=[0, 27, 25, 20, 4, 6, 15, 28, 9, 17, 24, 23, 22, 26, 12, 8, 10, 29, 1, 2, 13, 16, 18, 21, 19, 14, 3, 11, 5, 7]

最短路径=5565.380000000001

6.1.2 收敛过程



6.2 时间花销

程序共运行完整的 GA 遗传算法 10 次，总耗时在 10 分钟左右，平均每次算法耗时约为 1 分钟。针对本问题的规模——途经点数=30、闭合环路数=5、种群所含个体数=100、遗传迭代次数=6000，该时间花销较为理想，可以评定为在较短时间内实现了解的快速收敛。

6.3 最终解的最优性

本文所采用的算法共有 2 处提高了最终解的最优性：首先，基因交叉创新采用了“随机双点交叉法”，保证了解的全局搜索性；其次，程序共运行了完整的 GA 遗传算法 10 次，通过提高尝试次数而提高逼近全局最优解的概率。

经实验验证，运行 100 次与运行 10 次完整 GA 遗传算法所求得的最终解十分相近，数值相差在 20 以内，误差率=0.3%，故可以评定为最终解收敛于全局最优解。

综上所述，本文所选用的 GA 遗传算法很好地求解了模拟问题，由问题的等效性可知，该算法亦适用于求解原问题，可以在较短的时间内得到全局最优性极佳的最终解，故原问题得解。

7. 未来研究方向

1）继续改良 GA 遗传算法

由于启发式算法最终解的最优性与初始可行解有较大关系，为了使初始可行解尽可能落在全局最优解邻域内，针对本问题，可尝试联合 K-means 聚类算法，构建初始种群，使其尽可能落在全局最优解邻域内，进一步提高最终解的最优性。

2）尝试使用蚁群算法

蚁群算法虽并不十分适合多目标规划问题，但并非完全不可为之，只是建模与求解较为复杂困难；由于其正反馈性质，采用蚁群算法很有可能会使解较早的收敛于全局最优解，其收敛性也许会比解的搜索性较为随机的遗传算法更优秀。

3）考虑容量限制，将问题建模为 CVRP 问题

本模型一个重要假设是“各运输车容量相同，各社区物资需求量相同，且每辆运输车容量恰好=6*各社区物资需求量”，并不十分贴近实际情况，较为理想化，未来可个性化定义各运输车容量、各社区需求量，将问题转化为“CVRP 带容量限制的车辆路径规划问题”，进行建模求解，更具现实意义。

致谢：

感谢段老师和郑老师一学期以来的教导，通过选修“运筹学”这门我培养方案外的课程我受益匪浅！

值此新年佳节之际，祝老师们新年快乐，阖家幸福，万事顺遂！

本论文和 python 程序几乎 100% 原创，故参考文献较少，还请老师理解~

参考文献：

[1] https://blog.csdn.net/WFRainn/article/details/80458246