旅行商问题

模型概述

旅行商问题（traveling salesman problem，TSP）是典型的NP-hard问题，在车辆路径规划上有着广泛应用。它可以描述为：任意给定n个城市和它们之间的两两直达距离，需要找一个旅行总距离最短的闭合回路，使得每个城市经过且只经过一次。

给定一个完全图$G=(V,E)$，用$c_{i,j}$表示边的权重（距离，成本等），用0-1变量$x_{i,j}$表示路径中是否含有顶点i到顶点j，这样该问题的目标是追求总花费最少，即

$$\min Z=\sum_{i\in V}\sum_{j\in V}c_{i,j}x_{i,j} \tag{1}$$

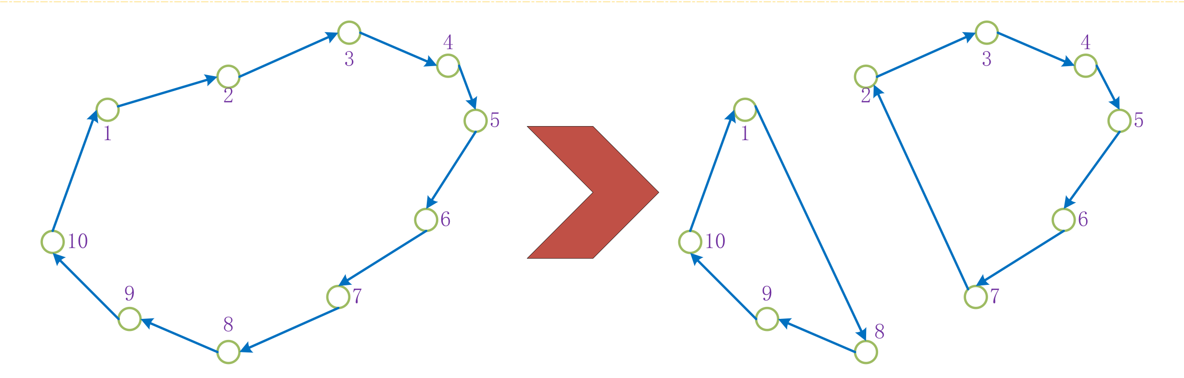
对于每一个顶点，流入该边的数量为1，即

$$\sum_{i\neq j}x_{i,j}=1,\forall j\in V \tag{2}$$

对于每一个顶点，流出该边的数量为1，即

$$\sum_{i\neq j}x_{i,j}=1,\forall i\in V \tag{3}$$

但是仅靠以上两个约束不足以得到需要的路径，因为可能出现子环路（subtour），如下图



对于消除子环路需要增加约束，后面有许多经典的模型。

算例1

已知有100个目标和基地的经纬度，有一架飞机从基地出发，侦察完所有目标，再返回原来的基地。在每一个目标点的侦察时间不计，求该架飞机所需要的最短距离。本算例为对称TSP。

距离矩阵

问题给出的是地理坐标，需要求两点的实际距离。设A，B两点的地理坐标分别为$(x_1,y_1),(x_2,y_2)$，过A，B两点的大圆的劣弧长几位两点的实际距离。以地心为坐标原点O，以赤道平面为XOY平面，以0度经线圈所在的平面为XOZ平面建立三维直角坐标系。则A，B两点的直角坐标分别为

$$A(R\cos x_1\cos y_1,R\sin x_1\cos y_1,R\sin y_1),B(R\cos x_2\cos y_2,R\sin x_2\cos y_2,R\sin y_2) \tag{4}$$

其中：$R=6370$千米为地球半径。那么AB的实际距离为

$$d=R\arccos\frac{OA\cdot OB}{|OA|\cdot|OB|} \tag{5}$$

$$= R \arccos\left[\cos\left(x_1 - x_2\right)\cos y_1 \cos y_2 + \sin y_1 \sin y_2\right]$$

显然，$d_{i,i} = 0, \forall i \in V$。

算例2

有130个城市需要巡游，给定X，Y坐标。

DFJ模型

设子环路S是所有节点集合V的一个真子集，即满足$S \subseteq V, 1 < |S| < |V|$，观察子环路可以发现其存在两边之和等于环路节点数，即有

$$\sum_{i,j \in S} x_{i,j} = |S| \tag{6}$$

对于$|S|$个顶点，形成回路至少要$|S|$条边，因此想要破除子环路，只需令

$$\sum_{i,j \in S} x_{i,j} \leq |S| - 1, 2 \leq |S| \leq |V| - 1, S \subseteq V \tag{7}$$

也可以使用使用等价描述为

$$\sum_{i \in S, j \notin S} x_{i,j} \geq 1, 2 \leq |S| \leq |V| - 1, S \subseteq V \tag{8}$$

总的模型可以写成以下这种形式

$$\begin{aligned}
&\min Z = \sum_{i \in V} \sum_{j \in V} c_{i,j} x_{i,j} \\
s.t.& \\
&\sum_{i=1}^{N} x_{i,j} - x_{j,j} = 1, j = 1, 2, \cdots, N \\
&\sum_{j=1}^{N} x_{i,j} - x_{i,i} = 1, i = 1, 2, \cdots, N \\
&\sum_{i,j \in S} x_{i,j} \leq |S| - 1, 2 \leq |S| \leq |V| - 1, S \subseteq V
\end{aligned} \tag{9}$$

这种约束是比较紧的约束，但(7)一共有$2^N - N - 1$个约束，当求解规模很大时难以枚举，有两种方式可以处理。

PlainLoop

采用一个大的while loop，先不添加任何子回路约束，然后在while loop中每次求解完成后判断是否产生了子回环，如果产生，那么添加对应的约束，重新求解整个模型，直至求得的解不包括子回路为止。

MTZ模型

引入辅助决策变量$\mu_i = 1, 2, \cdots N, i \in V$，这个决策变量可以认为是标签，用来标记顺序，以达到消除所有圈（包括最大圈）的目的。

对于每条边，构造约束

$$\mu_i - \mu_j + M x_{i,j} \le M - 1 \tag{10}$$

为了收缩上界，取$M = N, N = |V|$即顶点数量。

当$x_{i,j} = 1$即解存在从顶点i到顶点j的路径时，(10)可以化简为

$$\mu_j - \mu_i \ge 1 \tag{11}$$

当$x_{i,j} = 0$即解不存在从顶点i到顶点j的路径时，(10)可以化简为

$$\mu_i - \mu_j \le N - 1 \tag{12}$$

从(11), (12)可以看出从解的开始顶点μ_1到结束顶点μ_N，标签每次至少增加了1，总共至少增加了$N-1$，而从开始到结束最多增加了$N-1$，由抽屉原理说明，每次都是增加了1。

如果存在任意环路，从任意顶点出发到最后一个顶点的标签都是增加，但到自身时不可能再增加，这便破除了子环路。但是这也破除了最大环路，因此破除子环路不能考虑起点。

为了方便编程计算，可以考虑通过式(10)来得到$x_{i,i} = 0, i = 1, 2, \cdots, N$，这样总的模型可以写成以下形式

$$\begin{aligned}
&\min Z = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{i,j} x_{i,j} \\
s.t.& \\
&\sum_{i=1}^{N} x_{i,j} = 1, j = 1, 2, \cdots, N \\
&\sum_{j=1}^{N} x_{i,j} = 1, i = 1, 2, \cdots, N \\
&\mu_i - \mu_j + (N-1) x_{i,j} \le N - 2, 1 < i, j \le N \\
&x_{i,j} \in \{0, 1\}, \mu_1 = 1, \mu_i = 2, \cdots, N
\end{aligned} \tag{13}$$

GG模型

引入虚拟变量$y_{i,j}, i, j \in V$表示在边$x_{i,j}, i, j \in V$上的流量，流量不能大于$N-1$，即

$$y_{i,j} \le (N-1) x_{i,j}, i, j \in V \tag{14}$$

每个顶点都需要一个货物，从第一个顶点出发携带$N-1$个货物，即

$$\sum_{j \in V} y_{1,j} = N - 1 \tag{15}$$

每经过一个顶点就放下一个货物，即

$$\sum_{i \in V} y_{i,j} - \sum_{k \in V} y_{j,k} = 1, \forall j \in V \backslash \{1\} \tag{16}$$

其中，(14)可以进一步被缩紧为

$$\begin{gathered}
y_{i,j} \le (N-1) x_{i,j}, i = 1, j \in V \backslash \{1\} \\
y_{i,j} \le (N-2) x_{i,j}, i, j \in V \backslash \{1\}
\end{gathered} \tag{17}$$

总的模型可以写为

$$\min Z = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{i,j} x_{i,j}$$

$s.t.$

$$
\begin{aligned}
s.t. \quad & \\
& \sum_{i=1}^{N} x_{i,j} - x_{j,j} = 1, j = 1, 2, \cdots, N \\
& \sum_{j=1}^{N} x_{i,j} - x_{i,i} = 1, i = 1, 2, \cdots, N \\
& \sum_{j=2}^{N} y_{1,j} = N - 1 \\
& \sum_{i=1}^{N} y_{i,j} - \sum_{k=1}^{N} y_{j,k} = 1, j = 2, 3, \cdots, N \\
& y_{1,j} \leq (N-1) x_{1,j}, j = 2, 3, \cdots, N \\
& y_{i,j} \leq (N-2) x_{i,j}, i, j = 2, 3, \cdots, N \\
& x_{i,j} \in \{0, 1\}, y_{i,j} \geq 0
\end{aligned}
\tag{18}
$$

整数规划比较

计算机使用Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz，拥有6个物理核心，12个逻辑处理器，最高12线程，环境为win64 - Windows 11+.0 (22635.2)。

数据为算例1，GUROBI版本为11.0.0。

Method	N	Incumbent	BestBound	Gap%	Runtime(seconds)
DFJ_PlainLoop	101	39240.39	39240.39	0	35.56
MTZ	101	39240.39	38879.07	0.29	300.2
SCM(GG)	101	39240.39	39240.39	0	18.86

数据为算例2，GUROBI版本为11.0.0。

Method	N	Incumbent	BestBound	Gap%	Runtime(seconds)
DFJ_PlainLoop	130	6042.23	6042.23	0	33.46
MTZ	130	6176.3	5963.8	3.44	300.4
SCM(GG)	130	6042.23	6042.23	0	85.02

数据为算例1，COPT版本7.0.6。

Method	N	BestSolution	BestBound	Gap%	Runtime(seconds)
DFJ_PlainLoop	101	39240.39	39240.39	0	37.18
MTZ	101	43882.16	38375.72	12.55	300.2
SCM(GG)	101	232433.79	38762.55	83.3	300.3

数据为算例2，COPT版本7.0.6。

Method	N	BestSolution	BestBound	Gap%	Runtime(seconds)
DFJ_PlainLoop	130	6042.23	6042.23	0	33.52
MTZ	130	45626.78	5720.56	87.46	300.4
SCM(GG)	130	43782.42	5853.83	86.63	300.4

改良圈算法

又叫2-opt算法，首先随机给定Hamilton 圈C， 然后适当修改C以得到具有较小权的另一个Hamilton 圈，这种修改的方法叫做改良圈算法。

设初始圈$C = v_1 v_2, \cdots, v_n$

对于$1 \leq i < i+1 < j \leq n$，构造新的Hamilton 圈
$C_{i,j} = v_1 v_2 \cdots v_i v_j v_{j-1} v_{j-2} \cdots v_{i+1} v_{j+1} v_{j+2} \cdots v_n$，它是由$C$中删去边$v_i v_{i+1}$和$v_j v_{j+1}$，添加边$v_i v_j$，和$v_{i+1} v_{j+1}$而得到的。若$w(v_i, v_{i+1}) + w(v_j, v_{j+1}) < w(v_i, v_j) + w(v_{i+1}, v_{j+1})$ 则以$C_{i,j}$代替C，称之为改良圈。

一直重复该步骤，直至无法改进。

这样，每一次的迭代次数不确定，但可以进行多轮，来得到更好的结果。

遗传算法

编码策略

采用十进制编码，用随机数列$\omega_1 \omega_2, \cdots, \omega_N$作为染色体，其中$0 \leq \omega_i < 1, i = 1, 2, \cdots, N$，每个染色体都和种群中的一个个体对应，例如9目标问题的一个染色体为
$[0.23, 0.82, 0.45, 0.74, 0.87, 0.11, 0.56, 0.69, 0.78]$。

其中，编码位置i表示目标i，位置i的随机数表示目标i在巡回中的顺序，将这些随机数按升序排列得到如下巡回$6, 1, 3, 7, 8, 4, 9, 2, 5$。

种群初始化

初始化阶段，采用随机方式生成M个初始个体$C = \pi_1, \pi_2, \cdots, \pi_N$。

目标函数

目标函数为总路程最小，适应度可取为目标函数的相反数，追求

$$\min f(\pi_1 \pi_2, \cdots, \pi_N) = \sum_{i=1}^{N-1} d_{\pi_i, \pi_{i+1}} + d_{\pi_N, \pi_1} \tag{19}$$

交叉

交叉操作将父代个体的基因代码进行重组后生成子代个体，每个子代个体都会从父代个体中继承一些性状特点。这里使用单点交叉，将两个个体的染色体的部分基因交换。交叉策略为全部参与，适应度相当的交叉，将适应度排序，大的和大的交叉。

例如，个体1染色体$(0.14, 0.25, 0.27, 0.29, 0.54, \ldots, 0.19)$，个体2染色体$(0.23, 0.44, 0.56, 0.74, 0.21, \ldots, 0.24)$，设交叉点为第四个基因处，则交叉过后，个体1染色体$(0.14, 0.25, 0.27, 0.74, 0.21, \ldots, 0.24)$，个体2染色体$0.23, 0.44, 0.56, 0.29, 0.54, ., 0.19$。

交叉位置使用混沌序列产生一个2到N的正整数，具体是取一个$(0,1)$区间上的随机数作为初始值，然后利用$x_{n+1}=4x_n(1-x_n)$迭代一次产生 1个$(0,1)$区间上的混沌值，保存以上混沌值作为产生下一代交叉项的混沌迭代初值，再把这个值分别乘以N并加上1，最后取整即可。

这种交叉方式对原来解的改动很小，有利于保留优良性状。

变异

变异也是实现群体多样性的一种手段，是跳出局部最优，全局寻优的重要保证。这里变异算子设计如下，首先根据给定的变异率，随机地取两个在2到101之间的整数，对这两个数对应位置的基因进行变异，变异时利用混沌序列把这两个位置的基因换成新的基因值，从而得到新的染色体。

选择

采用确定性的选择策略，也就是说在父代种群和子代种群中选择目标函数值最小的M个个体进化到下一代，这样可以保证父代的优良特性被保存下来。

种群迁徙

为了增加基因多样性，每次迭代，就加入$0.1M$个初始化个体，然后选择淘汰适应度低的个体。

结果

将上述过程不断重复，如果最优个体连续20代不变化，种群就达到最优，停止循环。最大进化代数G。

结果比较

对算例1

Method	PopulationSize	MaxIteration	BestSolution	Runtime(seconds)
CM	50	100	42217.3	46.55
GA_Own	500	5000	102930.74	79.97
GA_SKO	200	2000	41297.19	62.97
GA_CM	20	40	42716.16	83.71

自己编写的GA算法在求解TSP问题时有一定的作用，但是收敛速度不如CM，如果将CM后的种群作为初始种群，那么GA算法很难提高目标值。