

第二次作业

题目 1. 用模拟退火算法求解城市数为 $N = 10$ 的 TSP 问题. 初始温度 $T_0 = 168.14$, 终止温度不大于 0.01, 城市坐标如下:

```

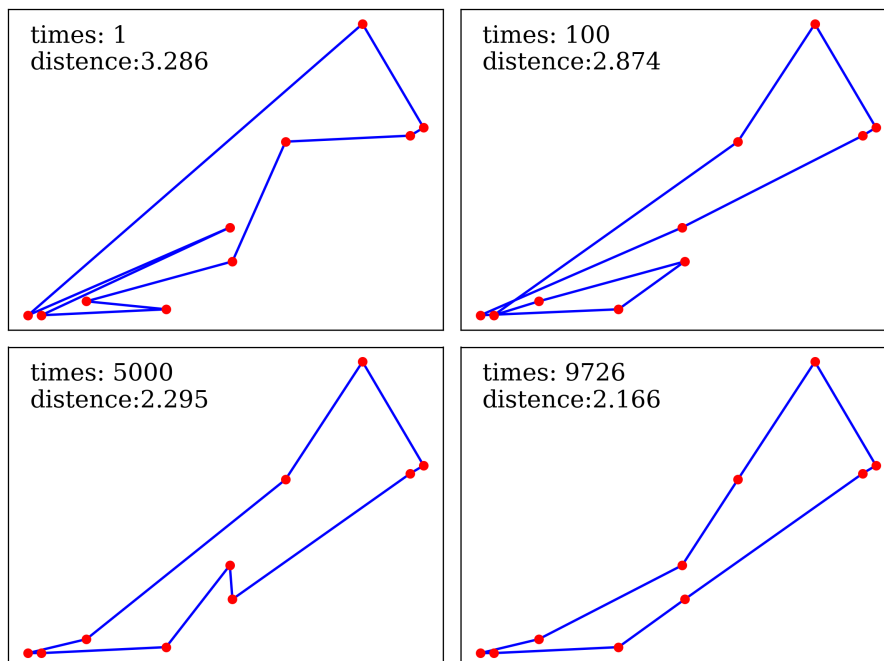
1 [[0.4    0.2439]
2  [0.1707 0.2293]
3  [0.5171 0.4439]
4  [0.1463 0.2293]
5  [0.761  0.9414]
6  [0.8732 0.6878]
7  [0.8488 0.6683]
8  [0.6195 0.6536]
9  [0.5219 0.3609]
10 [0.2536 0.2634]]

```

解答. 根据 Metropolis 准则, 对于第 t 次随机结果 E_t , 温度为 T_t , 从 E_{t-1} 转移到 E_t 的概率大小为

$$\exp \left\{ -\frac{E_t - E_{t-1}}{K \cdot T_t} \right\}$$

其中 K 为 Boltzmann 常数. 下面给出 $K = 1$, 衰减系数为 $\delta = 0.999$ 时的 TSP 问题求解效果:



固定初始温度, 终止温度, 衰减系数不变, 修改 $K = 1, 5, 10, 100$ 可得最终距离和 K 的关系如下:

| K | 1 | 5 | 50 | 100 |
|------|-------|-------|-------|-------|
| 最优距离 | 2.166 | 2.166 | 2.183 | 2.170 |

完整代码如下：

```
1 import numpy as np
2 from pathlib import Path
3 import matplotlib.pyplot as plt
4 plt.rcParams['font.family'] = ['serif']
5 PATH_FIGURE = Path(__file__).parent
6
7 np.random.seed(42)
8 rand = np.random.rand; randint = np.random.randint # 重载随机函数名
9 citys = np.loadtxt(Path(__file__).parent.joinpath("data.txt")) # 数据读入
10 n = citys.shape[0]
11 def getdis(path):
12     dis = 0;
13     for i in range(n):
14         dis += np.sqrt(np.sum(np.power(citys[path[i]] - citys[path[(i+1)%n]],
15                                     ↪ 2)))
16     return dis
17 def plot(path, cnt, ax:plt.Axes):
18     for i in range(n):
19         x, y = citys[path[i],:], citys[path[(i+1)%n],:]
20         ax.plot([x[0],y[0]], [x[1],y[1]], '-b')
21     ax.plot(citys[:,0], citys[:,1], '.r', markersize=10)
22     ax.text(0.15, 0.83, f"times: {cnt}\ndistance:{getdis(path):.4}", fontsize=15)
23     ax.set_xticks([]); ax.set_yticks([])
24
25 fig, axs = plt.subplots(2, 2, figsize=(8, 6)); ax = iter(axs.reshape(-1))
26 T = 168.14; K = 1; delta = 0.999; cnt = 0
27 path = list(range(n)); now = getdis(path)
28 best = {'path': path.copy(), 'dis': now}
29 while T >= 0.01:
30     r = rand(); path_ = path.copy()
31     if r <= 1/3:
32         a, b = randint(0, n, size=2) # 交换 a 和 b
33         while b == a: b = randint(0, n)
34         path[a], path[b] = path[b], path[a]
35     elif r <= 2/3: # 使 [s,s+1) 区间段逆序
36         s, l = randint(0,n,size=2); l = min(l, n-s); path[s:s+1] =
37             ↪ reversed(path[s:s+1])
38     else: # 将 path[p] 插入到位置 q 的前面
39         p, q = randint(0,n,size=2); idx = path[p]
40         path = path[0:p] + path[p+1:]
41         path = path[0:q] + [idx] + path[q:]
42     new = getdis(path)
43     if np.exp((now - new)/(T*K)) > rand(): now = new
44     else: path = path_
45     if now < best['dis']: best['dis'] = now; best['path'] = path.copy()
46     T *= delta; cnt += 1
47     if cnt in [1, 100, 5000]: plot(best['path'], cnt, next(ax))
48 plot(best['path'], cnt, next(ax))
49 print(f" 最优路径: {best['path']}, 长度: {best['dis']}")
50 fig.tight_layout()
51 fig.savefig(PATH_FIGURE.joinpath("TSP.png"), dpi=300); plt.show()
```
