

旅行售货员问题可视化项目

项目简介

本项目实现了旅行售货员问题(TSP)的两种求解算法：回溯法和分支限界法，并生成动画展示两种算法的搜索过程和性能对比。

环境要求

- Python 3.7+
- 推荐使用虚拟环境

依赖安装

```
pip install -r requirements.txt
```

主要依赖包：

- numpy：数值计算
- matplotlib：图形绘制
- Pillow (PIL)：图像处理和GIF生成

运行方式

一键运行（推荐）

直接运行主程序，将自动生成动画：

```
python tsp_visualization.py
```

程序将：

1. 使用预设随机种子(20009)生成8个城市的完全图
2. 分别用回溯法和分支限界法求解TSP

3. 生成动画文件 `tsp_animation.gif`
4. 在控制台输出求解结果和统计信息

自定义参数

可以修改 `tsp_visualization.py` 文件开头的配置常量：

```
NUM_CITIES = 8          # 城市数量（建议8-15）  
RANDOM_SEED = 20009     # 随机种子（用于复现）  
COORD_RANGE = 100.0     # 坐标范围  
MAX_EVENTS_PER_ALGO = float('inf') # 记录的搜索事件数量
```

输出说明

控制台输出

```
随机种子: 20009  
回溯法: 最优长度 = 268.45, 探索节点数 = 5880  
分支限界法: 最优长度 = 268.45, 探索节点数 = 735  
Animation saved to ./tsp_animation.gif (2095 frames)
```

动画文件

- **文件名:** `tsp_animation.gif`
- **位置:** 当前目录
- **内容:**
 - 图结构展示
 - 回溯法搜索过程
 - 分支限界法搜索过程
 - 两种算法结果对比
- **时长:** 约20-60秒（取决于记录的事件数量）
- **分辨率:** 1200×1200像素（满足≥720p要求）

项目结构

```
.  
├── tsp_visualization.py      # 主程序（算法实现+动画生成）  
├── requirements.txt          # Python依赖列表  
├── README.md                 # 本文件  
├── report.pdf                # 实验报告  
└── tsp_animation.gif         # 生成的动画文件（运行后生成）
```

数据生成

城市坐标在 $[0, 100]^2$ 范围内均匀随机生成，距离为欧几里得距离。使用固定随机种子保证可复现性。

随机种子

默认种子： 20009

可通过修改 `RANDOM_SEED` 变量更改，设为 `None` 将生成新的随机种子。

注意事项

1. **城市数量**：建议不超过8，过大会导致MemError(事实上n=8是已经爆内存，采取截断处理)
2. **内存占用**：动画生成过程中会占用较多内存
3. **运行时间**：8个城市约需5-6分钟，时间随城市数量指数增长
4. **GIF大小**：生成的GIF文件约20-30MB

故障排除

中文显示问题

如果动画中中文显示为方框，请确保系统安装了中文字体：

- Windows：通常已安装微软雅黑
- Linux：安装 `fonts-noto-cjk` 或 `fonts-wqy-zenhei`
- macOS：系统自带中文字体支持

内存不足

如果遇到内存错误，可以：

1. 减小 `MAX_EVENTS_PER_ALGO` 的值
2. 减少城市数量
3. 调整 `SMOOTH_FRAMES` 和 `PAUSE_FRAMES` 减少帧数