

智能优化算法及其应用

求解 TSP

自 66 夏卓凡 2016011496

2019 年 5 月 4 日

1 模拟退火算法与 TSP

本作业将使用模拟退火算法求解 TSP。模拟退火算法的步骤为，设定初始温度，从当前状态得到新的解，评价新的解，比当前解好直接接受，比当前解差按概率 $p_T = \exp(-\Delta t/T)$ 接受，在同一温度下重复上面的过程直到达到平衡，之后按一定规律降温，直到温度低于终止温度结束。模拟退火算法的流程图如下所示。

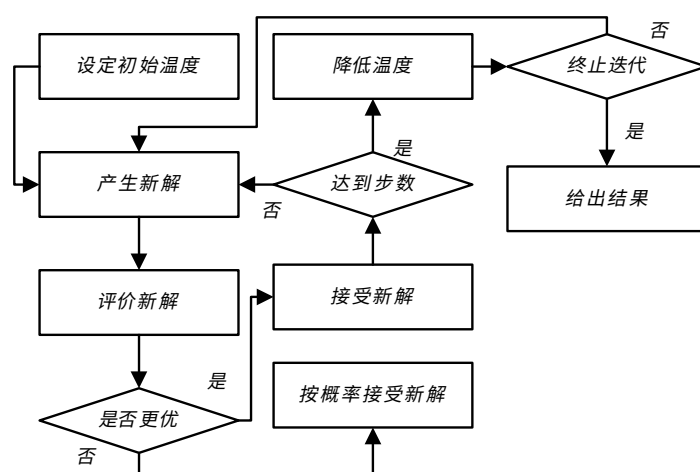


图 1: 模拟退火算法流程

TSP 即旅行商问题，是典型的 NP 完全问题，使用穷举法求解需要 $\mathcal{O}(n!)$ 的时间复杂度，无法满足实际需求。使用模拟退火算法这一智能优化算法可以在有限的时间内给出可以接受的较优解，这其中重点在于解的产生方式。假设给定 N 个城市的一个排列为 $\{x_1, x_2, \dots, x_n\}$ ，可以采取相邻两个元素交换位置，两元素之间序列逆置以及将一段序列移动到另一位置三种产生解的方式来构造新的解。本实验采用第二种方法，即序列逆置，实现的代码参考了博客https://blog.csdn.net/your_answer/article/details/79250624。TSP 的问题构造采取了均匀分布抽样的方法，在构造的问题中模拟退火算法表现良好。

2 求解结果分析

接下来给出城市数为 10, 30, 50 的一次实验结果。10 个城市时, 初始路径长度为 591.7737, 结束路径长度为 296.3357; 30 个城市时, 初始路径长度为 1742.6638, 结束路径长度为 512.6595; 50 个城市时, 初始路径长度为 2553.9111, 结束路径长度为 609.3898。

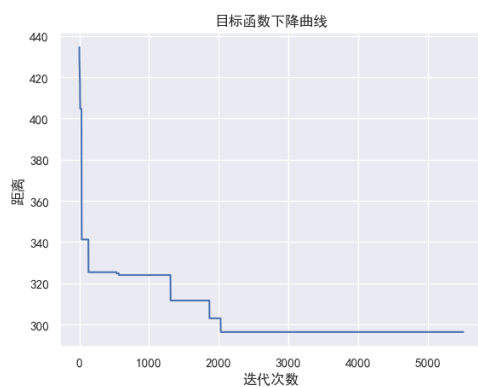


图 2: 10 个城市的目标函数变化

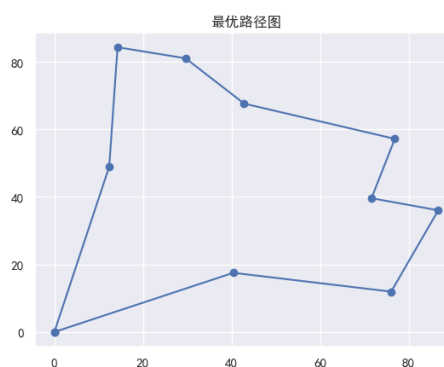


图 3: 10 个城市的结果图形

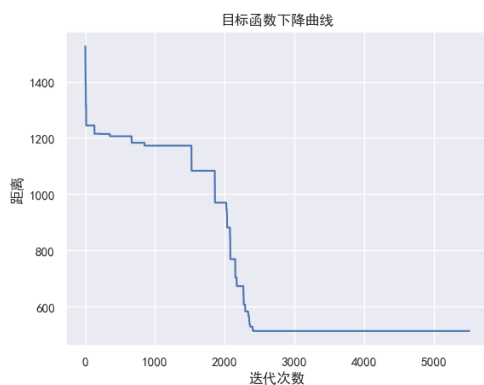


图 4: 30 个城市的目标函数变化

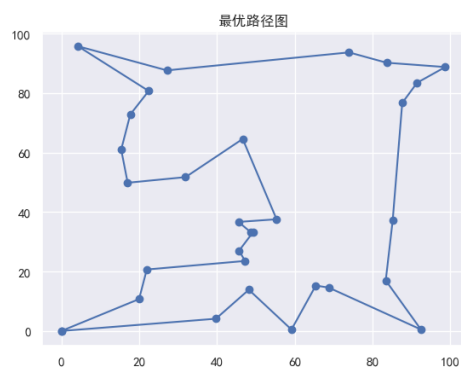


图 5: 30 个城市的结果图形

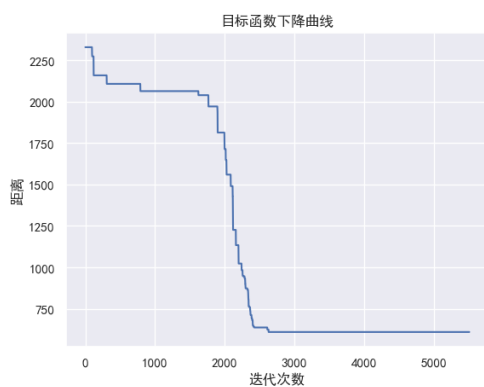


图 6: 50 个城市的目标函数变化

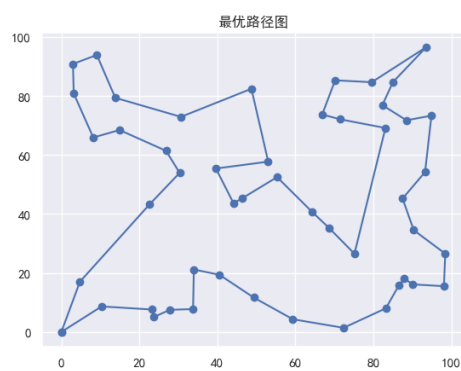


图 7: 50 个城市的结果图形

接下来对 20 次实验的表现进行分析, 列表如下。可以看出模拟退火算法的收敛性能比较稳定,

平均性能与最佳性能相差不大。

城市数	最佳性能	最差性能	平均性能	表现标准差
10	363.464331	363.464331	363.464331	0.000000
30	479.691389	494.782160	482.332297	4.747061
50	586.260879	630.027862	605.169167	9.513913

表 1: 20 次实验的统计结果

3 实验心得与体会

本次实验我尝试自己编写模拟退火但没能得到理想的结果，无奈只能参考网上的代码，实在很遗憾。事实上我一开始以为用模拟退火求解 TSP 运行时间会非常长，但实际上非常快，这说明作为一种智能优化算法，模拟退火确实比传统算法有极大的优势。

本次作业的代码的使用方法如下，各个超参数可以在代码内部设定。求解 10 城市的 TSP 需要指定`--num_city` 为 10。

```
1 python saa.py --num_city 10
```