基于模拟退火算法解决CVPR问题

陈昱丞 (学号: 1320231100)

**摘 要:**车辆路径优化问题是供应链优化中重要的环节。此CVPR问题中，目标是实现在约束条件下的路径最优，使得车辆组的总行驶距离最小。本问题中的约束条件有四：1.车辆数量固定；2.车辆容量固定；3.每个客户仅能由一辆车配送一次；4.车辆在完成配送任务后必须返回配送中心。此文使用模拟退火算法，运用交换算子和逆转算子两种邻域生成算子，采用线性退火和指数退火相结合的方式降温，避免在低温状态下的扰动较少问题。编码采用客户编号的排序，并设计其解码方法把约束考虑纳入其中。对于车辆数量固定的约束，采用惩罚机制，即若车辆安排超过固定数目，对目标函数加上一个较大惩罚项。实验中所给的实例的最优解如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实例 | 最优解 | 运行时间 | 实例 | 最优解 | 运行时间 |
| Instance 0 | 451 | 2.12 | Instance 3 | 1910 | 24.00 |
| Instance 1 | 808 | 3.53 | Instance 4 | 754 | 6.12 |
| Instance 2 | 1055 | 7.46 | Instance 5 | 1548 | 13.06 |

**关键词:** **CVPR 模拟退火 惩罚机制**

# 引言

## 问题背景介绍

在现代物流与供应链管理中，高效的货物配送是企业的核心竞争要素之一。无论是电商平台的快递配送、零售商的商品补货，还是城市垃圾清运、公共交通调度，都需要合理规划车辆的行驶路线，以提高服务效率。 车辆路径问题（Vehicle Routing Problem, VRP）是一类经典的组合优化问题，涉及如何有效地分配一组车辆去访问多个客户点，并在满足约束条件的情况下最小化总行驶距离。VRP在物流、配送、运输等领域有广泛的应用。

本问题的约束条件有四：1.车辆数量固定；2.车辆容量固定；3.每个客户仅能由一辆车配送一次；4.车辆在完成配送任务后必须返回配送中心。

## 问题建模

### 符号

|  |  |
| --- | --- |
| 符号 | 意义 |
|  | 客户顺序的一个序列 |
|  | 的不相容子序列加上depot节点，构成一个解 |
|  | 的不相容子序列 |
|  | 车辆经过边的距离 |
|  | 客户的需求量 |
|  | 火车最大载量 |
|  | 货车总数量 |

### 问题模型

定义为客户的一个排列，将拆开分为M个互不相容的子序列，并加上节点0，可构成一组问题的解。

式(1-1)去掉首位的0得到

式(1-2)满足



表示不同车辆经过的节点没有交集。

那么M辆车走过的子列的全程距离z有



对于某一条路径要满足客户需求不超过



其中，代表节点j的需求量。综上，有模型如下



## 解决方案

解决方案采用基本模拟退火算法的框架，解空间采用编码、解码的方式生成，针对车辆数目固定约束，若在优化过程中，规划车辆数大于固定数目，则对目标函数增加惩罚项。

本文后续部分组织如下。第2节详细陈述使用的方法，第3节报告实验结果，第4节对讨论本文的方法并总结全文。

# 算法设计

## 算法思路简介

**物理退火流程如下**：

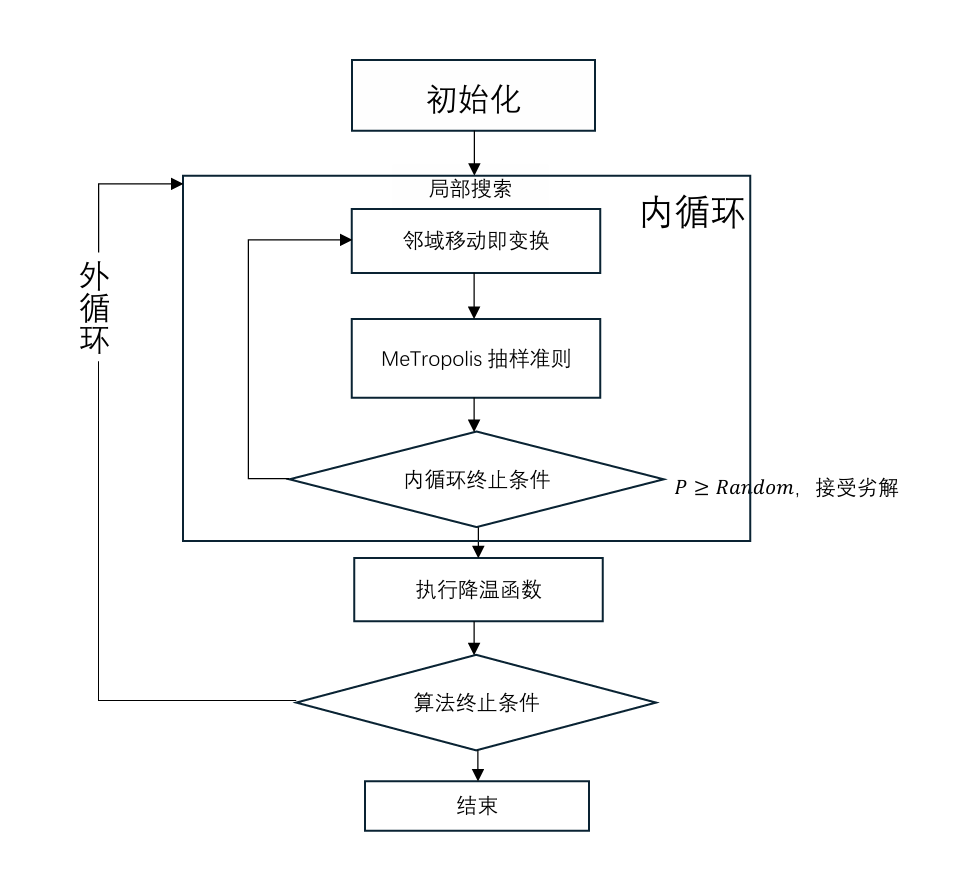
**高温：**粒子动能大，试图跳出当前状态。**等温：**不断跳跃到其他状态，粒子趋向动态平衡的运动。**降温：**粒子动能减小，去向稳定态。

**将其映射到最优化算法当中：**模拟退火算法就是将目标函数作为能量函数，当能量函数达到最小，目标函数也最小。**高温：**在初始解附近的解空间中局部搜索，搜索范围大，搜索次数多，接受新的劣解概率高。**等温：**在解空间中寻找新解，并根据准则判断是否接受新解。**降温：**局部搜索范围变小，接受新的劣解概率降低。

## 算法流程

**模拟退火算法流程**：如图(2-1)，按照Metropolis抽样准则接受新解，如式(2-1)





**(2-1)** 模拟退火算法流程图

**算法伪代码：**

|  |  |
| --- | --- |
| **算法** 模拟退火算法 | |
| **输入：**节点id数组，节点坐标数组，客户需求数组，货车载量，货车数量 | |
| **输出：**routes车辆路径数组，obj最终距离 | |
|  | **procedure** () |
|  | sol |
|  | **while** Tkiter **do** |
|  | **for**  **do** |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **if < 0 or >** random **then** |
|  | Sol |
|  | **end if** |
|  | **if < then** |
|  | Sol new\_sol |
|  | sol |
|  | **end if** |
|  | **end for** |
|  | **if then** Tk Tk |
|  | **end if** |
|  | **if then** Tk Tk |
|  | **end if** |
|  | **end while** |
|  | **end procedure** |

其中产生的邻域算子，参考文献[1] 我们选取交换算子和逆转算子

采用两种降温方式，线性降温和指数降温如下



## 时间复杂度和空间复杂度的分析

|  |  |  |
| --- | --- | --- |
| **算法模块** | **时间复杂度** | **空间复杂度** |
| 初始解生成 |  |  |
| 邻域生成 |  |  |
| 单词扰动 |  |  |
| SA总体 |  |  |

# 实验

## 实验设置

### 实验环境

实验使用 Python 3.12(anaconda) 在 Windows 11 操作系统上运行，主要依赖库包括NumPy和Matplotlib，代码使用 PyCharm 2023 社区版进行开发和调试。

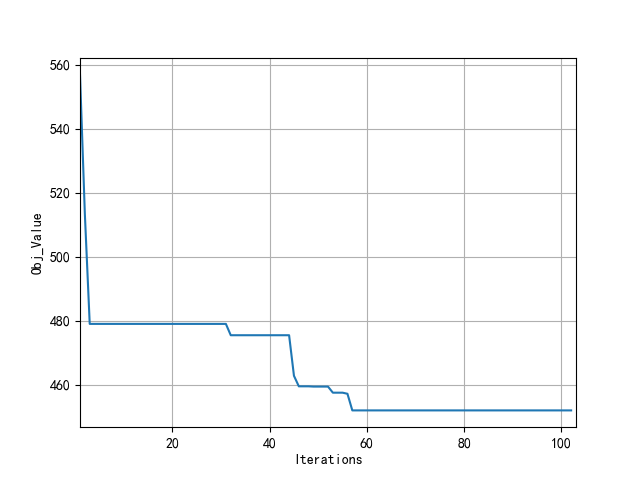
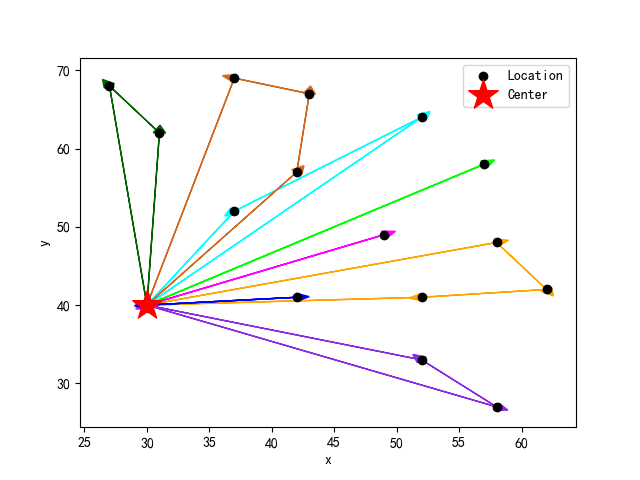
### 运行时间

运行时间如摘要结果所示

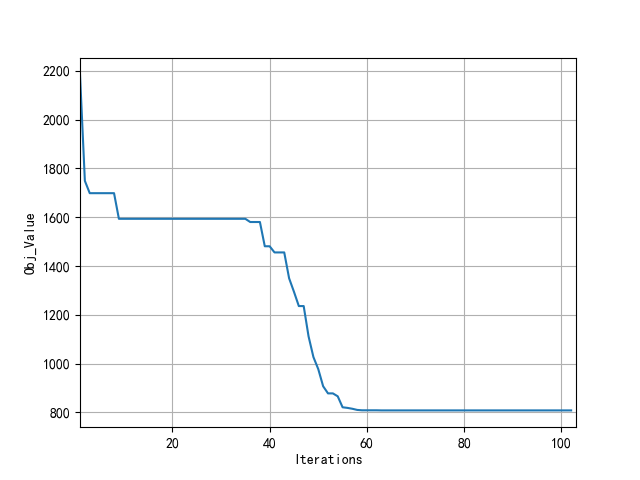
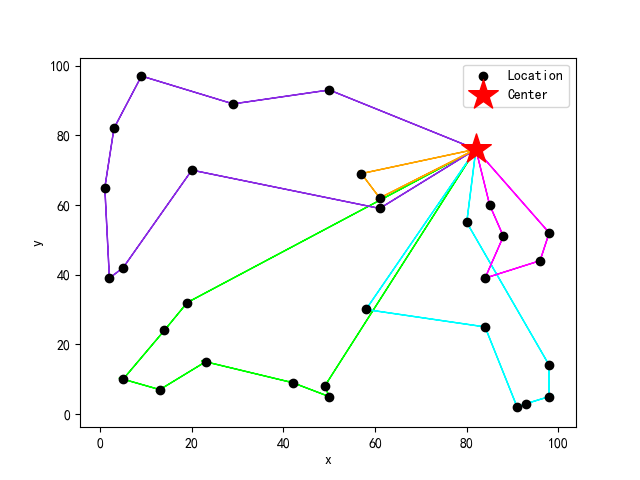
### 实验参数见3.2.2

## 实验结果 [可包含参数实验、对比实验等，可以组织在不同的小节中]

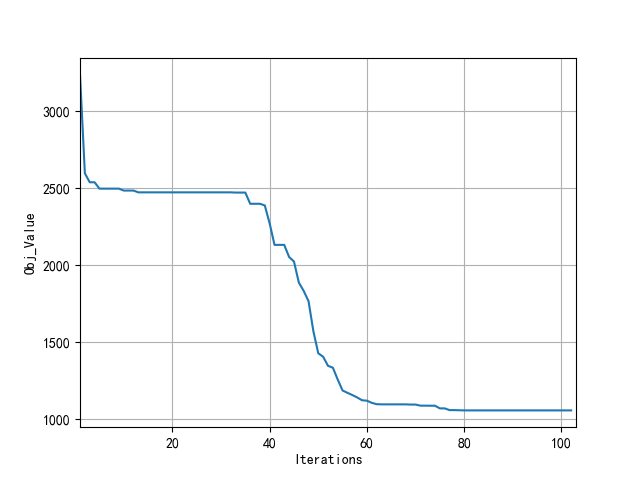
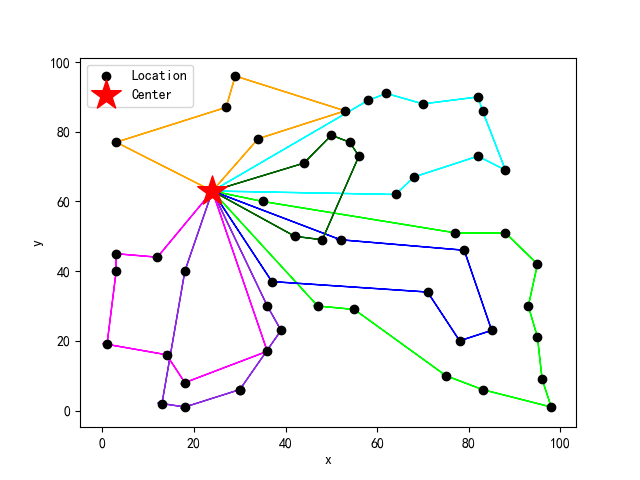
### 实验结果(T0=500, )



结果1 Instance 0



结果2 Instance 1



结果3 Instance 2

***为节省空间，Instance 3-5就不在此一一列出，可见文件夹/Output/内。***

左图为不同车辆的运行轨迹图，x轴、y轴为地理坐标轴。右图为目标函数随迭代次数的变化图。

### 依据经验的参数优化实验

选择参数关于如下

如下为在几组参数下对Instance 3的优化结果：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Instance 3的结果 |  |  |  | Instance 3的结果 |
| 500 | 0.8 | 0.01 | 1903.439 | 500 | 0.95 | 0.01 | 1846.671 |
| 300 | 0.8 | 0.01 | 1874.088 | 300 | 0.95 | 0.01 | 1823.110 |
| 200 | 0.8 | 0.01 | 1911.206 | 500 | 0.8 | 0.001 | 1903.439 |
| 500 | 0.9 | 0.01 | 1838.493 | 500 | 0.8 | 0.005 | 1903.439 |

**表3** 经验参数优化

可见，对于CVPR问题，当温度小于1时，降温幅度对最优解的影响较大，当变化时结果变化较为明显，且出现了更优解。这是因为更接近1的会使得降温收敛更慢，以更大几率找到局部最优解，即解决客户节点较为密集的问题。同时，初始温度对解的寻找影响较大，合适的初始温度更有利于寻找全局最优解，避免当温度下降而跳出全局最优解，陷入局部最优解的情况。

# 总结

本文针对具有固定车辆数量、固定容量等多重约束的CVPR（Capacity-constrained Vehicle Routing Problem）问题，设计并实现了一种改进的模拟退火算法。在算法设计上，采用了交换算子与逆转算子作为邻域生成机制，结合线性与指数两种降温策略，以提升搜索能力与收敛效率。同时，为适配车辆数量固定这一硬性约束，引入了惩罚机制，将超出车辆数目的解显著惩罚，从而引导搜索方向。

实验结果表明，该方法在多个典型实例上均得到了较优解，验证了所提方法的有效性与实用性。通过参数调优实验进一步发现，初始温度与降温系数对搜索性能有显著影响，适当选择参数能够提升最终解质量并加速收敛。

尽管本文方法已在一定程度上取得了良好效果，但仍存在以下不足：(1) 解的质量对初始解的依赖较强，未来可引入更优的启发式初始化策略；(2) 邻域结构相对简单，后续可尝试多种混合策略以增强局部搜索能力；(3) 算法参数仍依赖经验设定，今后可考虑引入自适应机制进行动态调节。

综上，本文所提出的方法为解决带有复杂约束的CVPR问题提供了一种可行的思路，并为后续更高效的路径优化算法设计提供了实践基础。

参考文献:

1. 吴艳群,董鹏.供应链中车辆路径问题的改进模拟退火算法[J].计算机工程与应用,2016,52(12):256-260.