

最优化方法大作业报告

王超艺

2024 年 1 月

1 问题背景与假设

1.1 背景介绍

某物流公司计划优化其配送网络，以最小化总配送成本，同时满足所有客户的需求和仓库的供给限制。配送成本与距离相关，每条配送线路的运输能力有限。配送网络不仅包括仓库与客户之间的直接配送路径，还允许仓库之间、客户之间相互配送，以实现更灵活和高效的物流调度。

1.2 主要假设

1. 节点定义：

- 仓库：共有 m 个仓库，且 $m \geq 5$
- 客户：共有 n 个客户，且 $n \geq 5$
- 网络节点集合：包含所有仓库和客户，共 $m+n$ 个节点

2. 配送路径：

- 路径类型
 - 仓库与仓库之间可以互相配送

- 客户与客户之间可以互相配送
- 仓库与客户之间可以直接配送。之所以考虑前两种路径，是因为仓库与客户之间的直接路径可能由于运输量有限，无法将全部的货物直接运输到对应的客户。

- 路径成本与距离成正比

3. **运输能力**：每条配送路径有运输能力上限

2 数学建模

2.1 决策变量

- 设 x_{kl} 表示从节点 k 到节点 l 配送的货物数量,其中 $k, l \in 1, 2, \dots, m+n$, 且 $k \neq l$
- 其中, 节点 1 到 m 为仓库, $m+1$ 到 $m+n$ 为客户

2.2 参数定义

- c_{ij} : 节点 i 到节点 l 的单位配送成本
- s_i : 仓库 i 的供应量 ($i=1,2,\dots,m$)
- d_j : 客户 j 的需求量 ($j=m+1,m+2,\dots,m+n$)
- u_{kl} : 节点 k 到节点 l 的运输能力上限

2.3 目标函数

最小化总配送成本：

$$\text{Minimize } Z = \sum_{k=1}^{m+n} \sum_{\substack{l=1 \\ l \neq k}}^{m+n} c_{kl} x_{kl}$$

2.4 约束条件

1. 供需平衡约束：对于每个节点 k

$$\sum_{\substack{l=1 \\ l \neq k}}^{m+n} x_{kl} - \sum_{\substack{l=1 \\ l \neq k}}^{m+n} x_{lk} = \begin{cases} s_k & \text{如果 } k \text{ 为仓库} \\ -d_k & \text{如果 } k \text{ 为客户} \end{cases} \quad \forall k = 1, 2, \dots, m+n$$

2. 运输能力限制：每条配送路径的货物数量不能超过其运输能力上限

$$x_{kl} \leq u_{kl} \quad \forall k, l = 1, 2, \dots, m+n; \quad k \neq l$$

3. 非负性约束

$$x_{kl} \geq 0 \quad \forall k, l = 1, 2, \dots, m+n; \quad k \neq l$$

3 算法选择

3.1 两阶段单纯形法

3.1.1 理论背景

在求解线性规划问题时，单纯形法（Simplex Method）是一种经典且高效的算法。然而，直接应用单纯形法往往需要初始可行解，而有些问题的初始可行解难以直接获得。为了解决这一问题，两阶段单纯形法（Two-Phase Simplex Method）被提出，用以在没有显式初始可行解的情况下求解线性规划问题。

两阶段单纯形法将线性规划问题的求解过程分为两个阶段：

1. 第一阶段：引入人工变量并最小化其总和
 - 当初始问题没有显式的可行解时，第一阶段的目标是找到一个可行解。为此，首先对每一个等式约束引入一个人工变量。

- 最小化目标函数，使所有人工变量取 0，从而获得原问题的可行解。

2. 第二阶段：基于第一阶段的可行解优化原目标函数

- 使用第一阶段得到的可行解作为初始基。
- 优化原始目标函数。通过单纯形法迭代，直到找到最优解。

3.1.2 选择理由

1. 问题的线性特性

- 物流配送网络优化问题的目标函数和约束条件均为线性的，因此，该问题天然适合使用线性规划方法求解，而两阶段单纯形法作为线性规划中的一种高效算法，能够有效处理此类问题。

2. 约束条件的满足

- 供需平衡约束：通过引入人工变量，确保每个节点（仓库或客户）的供需平衡得到严格满足。
- 运输能力限制：单纯形法在迭代过程中自动考虑所有不等式约束，确保每条配送路径的货物量不超过其运输能力上限。
- 非负性约束：单纯形法本身在搜索过程中维持所有决策变量为非负，符合物流配送问题的实际要求。

3. 全局最优解的取得

- 全局最优性：无论初始解如何，两阶段单纯形法都能通过线性规划的性质找到全局最优解，而非局部最优解。
- 效率高：尽管单纯形法的最坏情况时间复杂度为指数级，但在实际应用中通常表现出多项式时间的效率，尤其适用于中小规模的物流配送网络优化问题。

3.2 遗传算法

3.2.1 理论背景

1. 基本概念与组成

- 种群 (Population): 遗传算法以一组潜在解决方案 (称为个体或染色体) 作为初始种群, 每个个体代表问题的一个可能解。
- 个体 (Individual): 种群中的每一个成员, 通常用二进制串、实数串或其他编码方式表示, 称为染色体 (Chromosome)。
- 基因 (Gene): 染色体中的基本单元, 代表解的一个参数或部分。
- 适应度函数 (Fitness Function): 用于评估个体质量的函数, 衡量其在当前问题中的优劣程度。适应度越高, 个体被选中繁殖的概率越大。
- 选择 (Selection): 根据适应度选择优秀个体进行繁殖, 常用的方法包括轮盘赌选择 (Roulette Wheel Selection)、锦标赛选择 (Tournament Selection) 等。
- 交叉 (Crossover): 通过交换两个父代个体的部分基因, 生成新的子代个体, 模拟生物遗传中的基因重组过程。常见的交叉方式有单点交叉、多点交叉和均匀交叉等。
- 变异 (Mutation): 以一定概率随机改变个体的某些基因, 增加种群的多样性, 防止算法陷入局部最优。变异操作模拟生物中的基因突变过程。
- 终止条件 (Termination Condition): 确定算法停止的标准, 通常包括达到预定的迭代次数、适应度达到某一阈值或种群适应度收敛等。

2. 算法的工作流程

- (a) 初始化种群: 随机生成初始种群, 确保种群多样性。

- (b) 评估适应度：计算每个个体的适应度值，评估其解决问题的优劣。
- (c) 选择操作：根据适应度选择个体作为父代，准备进行交叉和繁殖。
- (d) 交叉操作：对选择的父代进行交叉，生成新一代的子代个体。
- (e) 变异操作：对部分子代个体进行变异，增加基因多样性。
- (f) 生成新种群：用子代替代旧种群，形成新的种群。
- (g) 终止判断：检查是否满足终止条件，如未满足则返回步骤 2，继续迭代。

3.2.2 选择理由

- 全局搜索能力强：通过种群并行搜索，具有较好的全局搜索能力，能够有效避免陷入局部最优。
- 适应性强：能够处理复杂的、多峰值的、多目标的优化问题，在合理时间内找到近似最优解。
- 灵活性高：可以结合多种编码方式和遗传操作，适应不同类型的问题。

3.3 模拟退火算法

3.3.1 理论背景

模拟退火算法是一种基于概率的全局优化算法，灵感来源于固体物理中的退火过程。模拟退火算法主要包括以下几个关键要素：

- 初始解 (Initial Solution)：选择一个初始解作为搜索的起点。
- 温度 (Temperature)：控制搜索过程的一个参数，决定了接受劣质解的概率。温度逐渐降低，模拟物理退火过程中的冷却。
- 冷却计划 (Cooling Schedule)：决定温度如何随迭代进行而逐步降低。常见的冷却计划包括指数衰减、线性衰减等。

- 邻域生成 (Neighborhood Generation)：在当前解的基础上，通过某种策略生成一个新的邻域解。策略可以是局部调整、交换等操作。
- 接受准则 (Acceptance Criterion)：根据目标函数的变化和当前温度，决定是否接受新的解。通常采用 Metropolis 准则。

模拟退火算法的基本步骤如下：

1. 初始化：设置初始温度 T_0 ，选择初始解 S_0 。
2. 迭代搜索：
 - 在当前解 S 的领域内产生一个新解 S'
 - 计算目标函数的变化 $\Delta E = f(S') - f(S)$ ，选择是否接收新解。
3. 冷却：根据冷却计划更新温度 T 。
4. 终止条件：当温度低于某一阈值或达到最大迭代次数时，终止算法。

3.3.2 选择理由

- 全局搜索能力强：通过接受劣质解，避免陷入局部最优，具有较强的全局优化能力。
- 实现简单：算法步骤简单，易于实现。
- 适用范围广：可用于各种复杂的优化问题，包括组合优化和连续优化。

4 实验与分析

4.1 实验数据

本次实验共设计了 8 个仓库，12 个客户。仓库与客户位置以及各仓库的供给量、客户的需求量如图 1 所示，其中距离单位为千米：

路径类型	随机范围 (吨)
仓库到客户	[10, 30]
仓库到仓库	[30, 60]
客户到客户	[50, 150]

表 1: 运输上限

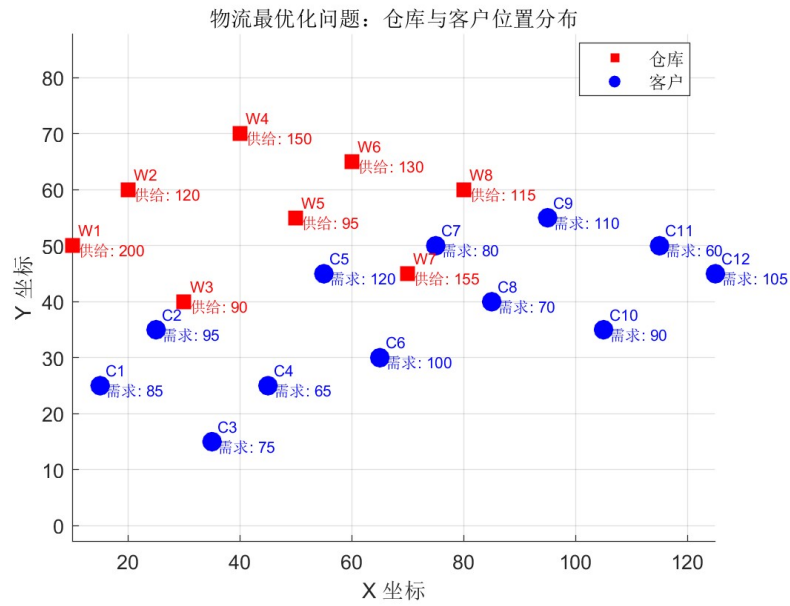


图 1: 实验数据

为了更好地模拟现实中多样的情况，节点之间路径的运输上限用随机数的方式生成。但是由于设置了固定的随机种子，所以每次生成的数据是一致的，因而可以做对比实验。表 1 是不同路径的运输上限随机范围。可以看到，仓库到客户的运输上限设置的比较小，这是为了模拟实际中的复杂情况，减少直接从仓库配送到目标客户的直接情形，迫使算法尝试将部分客户作为中转节点，进行第二次配送。并且对于任意 2 个节点 i 、 j ，从 i 到 j 和从 j 到 i 的运输上限相同。

4.2 最优配送方案总成本

算法名称	最优配送方案总成本 (元)
二阶段单纯形法	82711
遗传算法	215051
模拟退火算法	181080

表 2: 总成本对比

4.3 运输路径图

两阶段单纯形法:

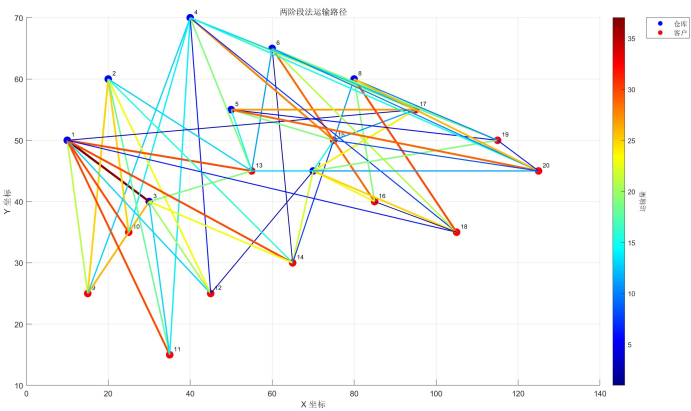


图 2: 两阶段单纯形法运输路径

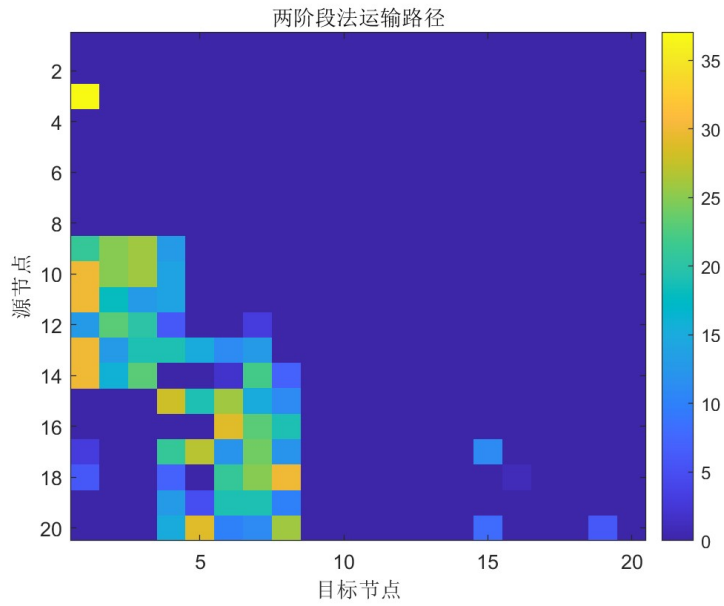


图 3: 两阶段单纯形法矩阵热图

由于遗传算法和模拟退火算法的运输路径过多，因而只绘制运输矩阵热图。在矩阵热图中，第 (i,j) 个格点代表从节点 i 到节点 j 的运输量。节点 1-8 代表仓库，节点 9-20 代表客户。

可以观察到，三种方法的右上角运输量都很小，这是因为从仓库到客户的直接线路，运输上限比较小。因而三种算法的整体趋势都是将货物按照较小的配额，分别运送到各个客户，然后以客户为中转节点，再次配送到最终的客户。

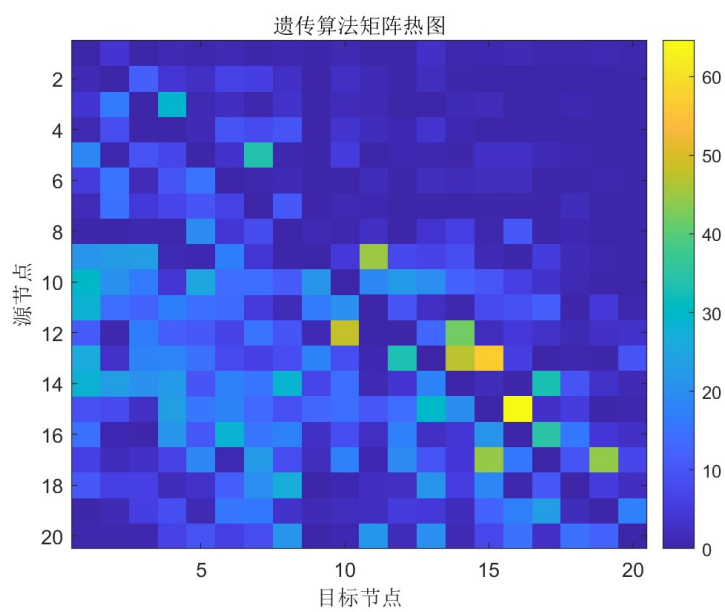


图 4: 遗传算法矩阵热图

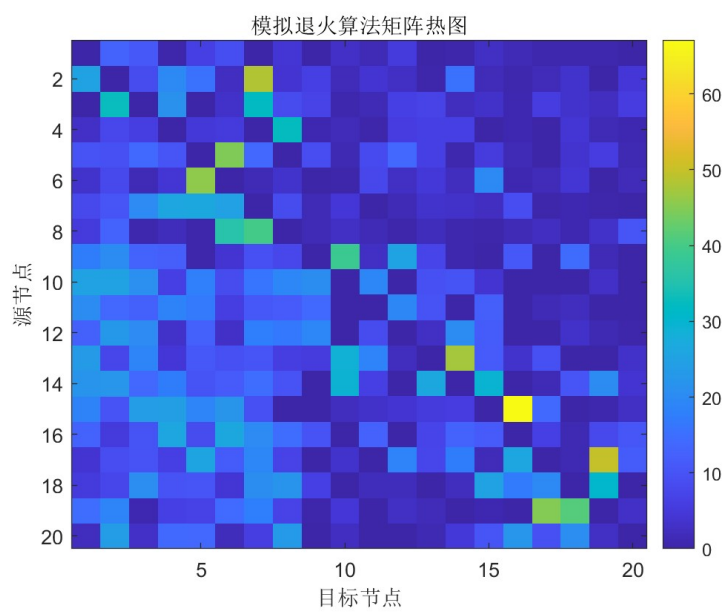


图 5: 模拟退火算法矩阵热图

4.4 约束满足情况

供需平衡满足情况 (纵坐标单位为吨)

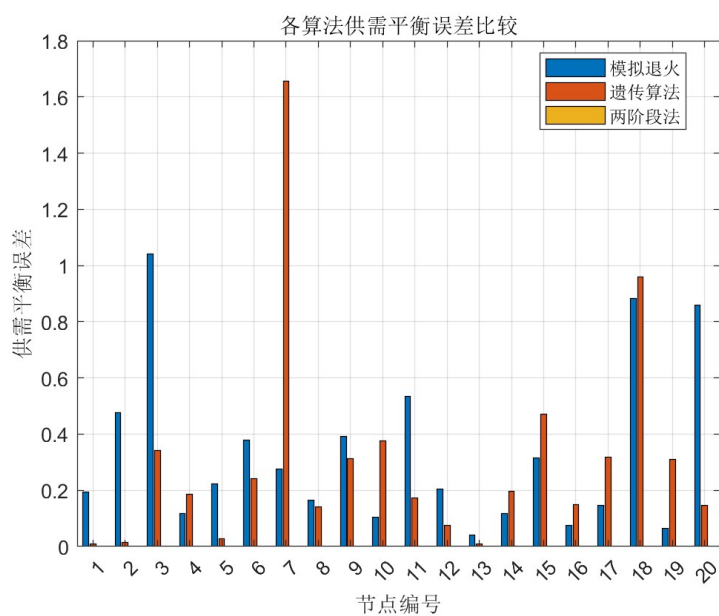


图 6: 各算法供需平衡误差比较

可以看到，由于约束条件的影响，二阶段单纯形法没有任何供需平衡误差，但是启发式算法都有一定的误差。

运输能力上限违背情况：经过计算验证，三种算法均未违背运输上限。

4.5 计算复杂度

算法名称	理论复杂度	运行时间 (s)
二阶段单纯形法	最坏情况下为 $O(2^n)$	2.8170
遗传算法	$O(G \times P \times (L + T_f))$	6.2670
模拟退火算法	粗略估计为 $O(K \times L)$	3.6325

表 3: 时间复杂度对比

其中, n 为二阶段单纯形法的变量个数; G 为遗传算法迭代次数, P 为个体数量, L 为个体长度, T_f 为适应度评估时间; K 为模拟退火算法迭代次数, L 为问题规模。

对于两种启发式算法, 表格中列出的运行时间为早停机制触发时的运行时间。但是为了更好地观察适应度曲线, 实际运行时还是在运行到指定的最大迭代次数后才停机。

4.6 算法收敛性能

两阶段单纯形法第一阶段迭代 540 次, 第二阶段迭代 424 次。

遗传算法的适应度曲线如下:

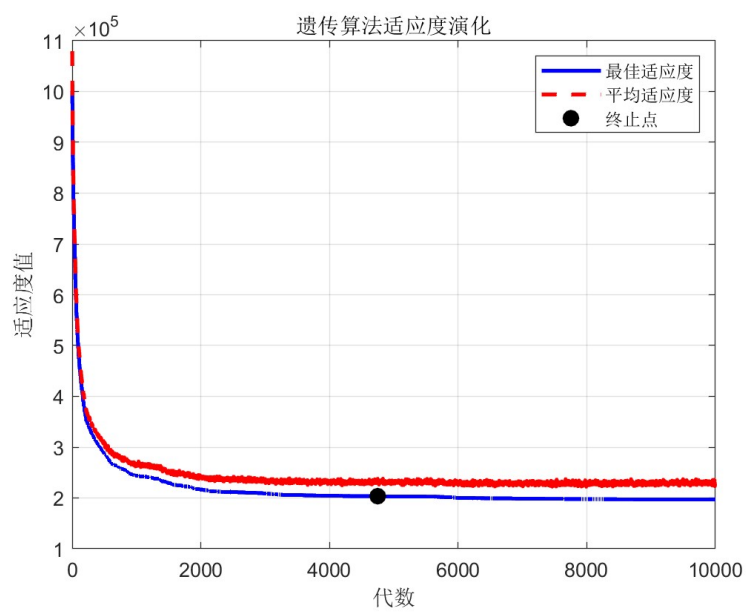


图 7: 遗传算法适应度曲线

模拟退火算法的适应度曲线如下：

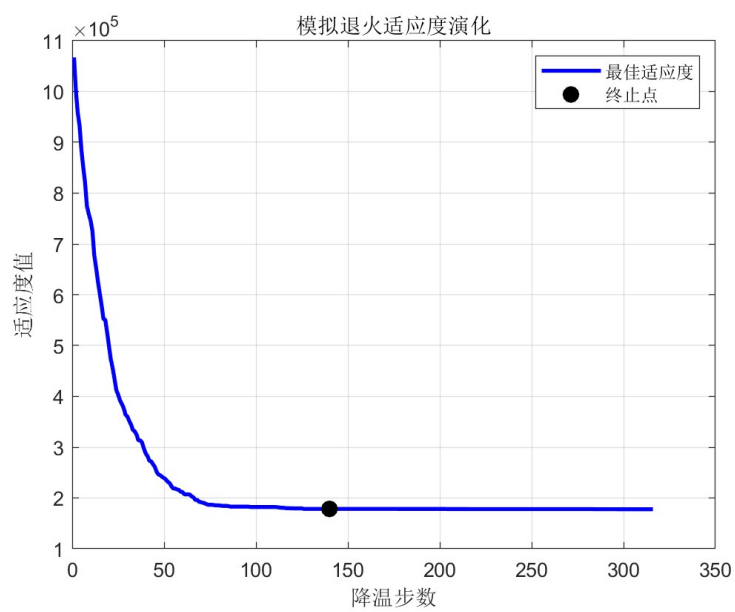


图 8: 模拟退火算法适应度曲线

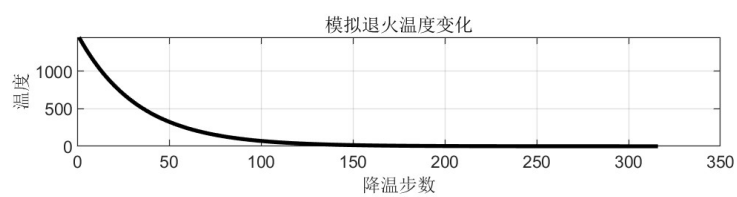


图 9: 模拟退火算法温度曲线

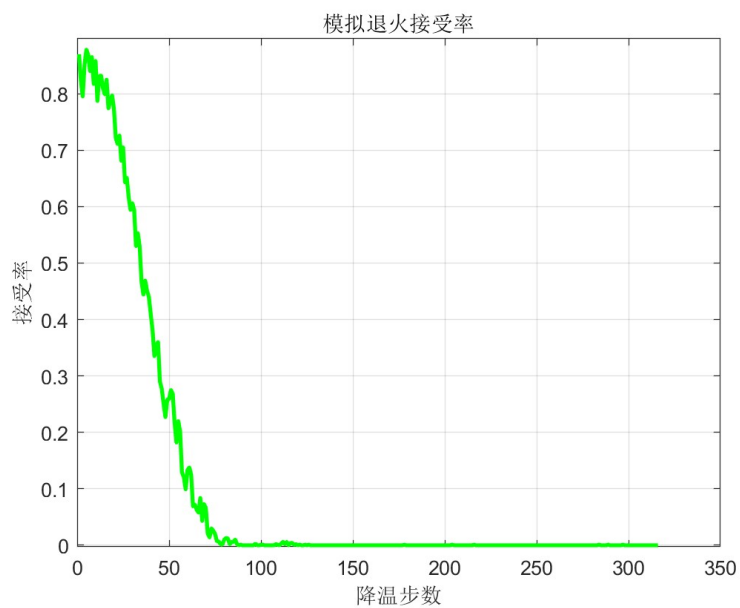


图 10: 模拟退火算法接受率曲线

可见，两种启发式方法都在一定迭代步数内达到了收敛。

4.7 结果分析

1. 概述

(a) 两阶段单纯形法

- 最优解：成功求得最优解。
- 约束满足：没有违反供应平衡约束。
- 运行速度：最快。
- 运输矩阵热图：分布不均匀，某些路径的运输量较大，而其他路径的运输量较小。

(b) 遗传算法和模拟退火算法

- 最优解：未能求得最优解。

- 约束满足：违反供应平衡约束。
- 运行速度：相对较长。
- 运输矩阵热图：分布较为均匀。

2. 原因分析

(a) 算法适用性

- 线性规划的适用性：两阶段单纯形法专为线性规划设计，能够高效处理具有线性目标函数和线性约束的问题。而 GA 和 SA 作为通用的启发式优化算法，虽然灵活性更高，但在处理线性问题时，缺乏单纯形法那样的专门优化机制，导致在精确性和效率上表现不佳。

(b) 约束处理机制

- 两阶段单纯形法通过引入人工变量和精确的线性代数操作，确保所有约束在求解过程中严格满足。
- 软约束处理：通常依赖罚函数或其他策略来处理约束，这可能导致部分约束被违反，尤其是在优化过程的早期阶段。虽然我对惩罚因子做了动态调整，但仍然无法保证不会违反约束。

(c) 搜索策略

i. 两阶段单纯形法

- 通过在解空间的顶点间移动，系统地优化目标函数，确保每一步都朝着最优方向前进。
- 通常会集中运输量在成本最低的路径上，导致某些路径运输量较大，而其他路径运输量较小。
- 这种集中运输策略有助于降低整体运输成本，但可能导致某些运输路线的资源利用率过高，潜在地引发运输瓶颈或资源过载。

ii. 遗传算法和模拟退火算法

- 依赖于随机性和概率性策略，探索更广泛的解空间，但缺乏系统性的优化路径，可能导致收敛速度较慢和精度不足。
- GA 和 SA 由于其全局搜索特性和多样性维护机制，倾向于在解空间中分散搜索，导致运输量分布更加均匀。
- 这种分布策略有助于平衡各运输路径的负载，减少单一路径的压力，但可能在成本上不如集中策略经济。

5 拓展与讨论

5.1 可能的复杂因素

1. 时间窗口 (Time Windows):

- 定义：每个客户在特定的时间段内可以接收货物。例如，客户 A 只能在上午 9 点到 11 点之间接收货物。
- 影响：增加了配送计划的时间维度，需要在满足客户需求的同时，遵守时间约束。

2. 动态需求 (Dynamic Demand):

- 定义：客户需求量随时间变化，可能基于历史数据或预测模型进行调整。
- 影响：使得需求不再是静态的，要求配送网络具备应对需求变化的灵活性。

3. 多目标优化 (Multi-objective Optimization):

- 定义：同时优化多个目标，如最小化成本、最小化配送时间、最大化客户满意度等。

- 影响: 增加了优化问题的复杂性, 需要在多个目标之间进行权衡。

5.2 改进方法

1. 混合整数线性规划 (Mixed-Integer Linear Programming, MILP):

- 引入整数变量以处理时间窗口和车辆路径选择等离散决策。
- 使用 MILP 模型将时间窗口和车辆容量等复杂约束纳入优化框架。
- 优点
 - 能够精确建模离散决策 (如车辆选择、路径选择)。
 - 可以在理论上保证全局最优解。
- 缺点
 - 计算复杂度高, 求解时间长, 尤其在大规模问题下难以应用。
 - 对于动态需求, 模型需要频繁重新求解, 计算成本高。

2. 启发式与元启发式算法 (Heuristic and Metaheuristic Algorithms):

- 结合遗传算法、模拟退火、粒子群优化等元启发式算法, 设计适应性强的优化策略。
- 采用分阶段优化或分解方法, 将复杂问题拆分为多个子问题逐步求解。
- 优点
 - 具备较强的适应性和灵活性, 能够处理复杂和大规模问题。
 - 可以在合理时间内找到近似最优解, 适用于实时或动态优化需求。
- 缺点
 - 无法保证全局最优解, 结果依赖于算法参数和初始条件。

- 需要复杂的调参，需要深入理解算法特性和问题结构。

3. 强化学习 (Reinforcement Learning, RL):

- 采用强化学习算法，训练智能体通过与环境的交互优化配送策略。
- 优点：能够处理高度动态和复杂的环境，具备学习和适应能力。
- 缺点
 - 需要大量训练数据和计算资源，训练过程复杂。
 - 结果的解释性较差，难以验证和调试。