

## 5 基于配送中心选址的优达生鲜配送路径优化

本章以哈尔滨新天地超市南岗区的实际配送问题为背景，针对生鲜产品配送对时效和质量严格要求，构建了考虑软时间窗约束的车辆路径问题（VRPTW）模型，并结合配送服务质量评价（采用层次分析法与模糊综合评价）以及需求预测，为配送中心选址与配送路径优化提供理论和方法支持。通过利用百度地图获取站点经纬度，并以南岗区中心坐标将经纬度转换为平面坐标，再结合改进的粒子群算法（PSO）进行求解，本文最终获得一组成本最低、距离最短且能够较好满足时间窗要求的配送方案。下文将从问题描述、模型建立、算法设计、模型求解与实例分析五个方面详细展开，并对结果进行深入讨论。

### 5.1 带软时间窗的 VRP 问题描述

在生鲜配送领域，由于产品对新鲜度和时效性要求极高，传统只考虑路径最短或载重约束的车辆路径问题（VRP）已无法完全反映实际配送中存在的各种限制。因此，本研究在传统 VRP 模型的基础上引入软时间窗约束，使得车辆在规定的时间内送达可获得低成本，反之则需承担等待或迟到的经济处罚，以反映实际运营中的时间敏感性。

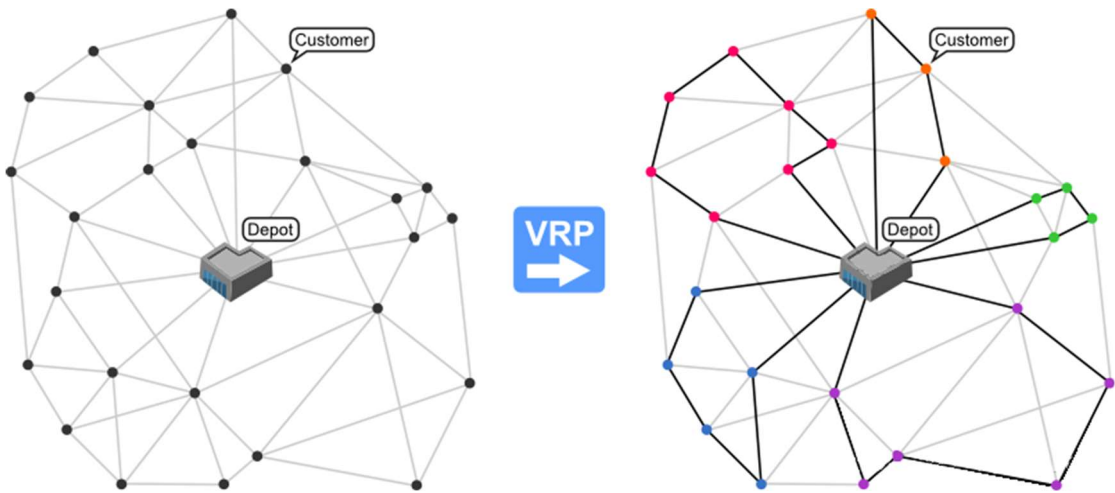


图 5-1 VRP 问题原理

#### 5.1.1 符号定义

为便于构建数学模型，本文采用下列符号和参数，并结合实验数据进行说明：

配送中心固定设在哈尔滨市道外区团结镇郭地方路 66-68 号传化公路港 B 区，记为节点 0；南岗区内新天地超市 14 个位置为客户节点，记为节点 1 至 14；

各节点的经纬度数据通过百度地图获取，经坐标拾取器校验后，通过 Python 程序转换为平面坐标，确保距离计算准确；

根据哈尔滨市南岗区实际路况的深入调研，本研究选择 4.2 米箱式货车作为配送车辆。南岗区作为哈尔滨市的核心商业区，道路狭窄且早晚高峰时段交通拥堵频发，大型货车通行受限。4.2 米车型兼具 5 吨载重能力与道路适应性，能满足单日配送需求同时又可规避因车型过大导致的通行效率下降问题。此车辆最大行驶距离（D）的设定综合考虑了城市路网密度与生鲜产品的保鲜周期，因为生鲜肉类在常温下保质期通常不超过 4 小时，车辆单次行驶距离过长时将导致末端配送节点的商品质量下降。所以通过与企业运营数据对比，最终将 D 设定为 100 公里以确保配送全程控制在 3 小时以内，平衡运输效率与商品损耗风险。

$q_i$ : 第  $i$  个客户的日均需求量，由区域人口密度、商业区辐射范围及历史销售数据推导；

$Q$ : 每辆车的最大载重，本研究中使用 4.2 米箱式车辆，载重约为 5t；

$D$ : 车辆的最大允许行驶距离，依据城市路网情况和配送范围设定；

$V$ : 车辆行驶速度，设定为 40 km/h，符合城市配送平均水平。

时间窗与服务参数：

每个客户节点  $i$  的服务时间窗记为  $[e_i, l_i]$ ，例如部分超市的时间窗设定大致范围为 8:00~12:00；

$s_i$ : 第  $i$  个客户节点的服务时长，反映配送员在超市进行卸货、签收等操作所需的时间。

$C_0$ : 车辆启动的固定成本，反映每辆车启程前的准备费用；

$C_1$ : 单位距离行驶成本，表示车辆行驶每公里所耗费用；

$C_2$ : 单位等待时间成本，若车辆提前到达客户处需要等待，则按此成本计费；

$C_3$ : 单位迟到时间成本，当车辆晚于最晚服务时间到达时产生，若车辆晚于规定时间到达则产生较高的惩罚费用。

其他决策变量与符号如下：

$x_{ij}$ : 二值变量，若车辆从节点  $i$  直接行驶到节点  $j$  则取 1，否则取 0；

$t_i^k$ : 表示车辆  $k$  到达节点  $i$  的实际时间；

若考虑多配送中心选址，则引入  $\delta_k$  表示候选配送中心是否被选中，本研究仅考虑单一

配送中心的情况。

### 5.1.2 目标函数

模型目标是 최소화 总配送成本，主要由以下部分组成：

#### 1. 车辆运行成本：

每辆车启程产生固定启动成本  $C_0$ ；

行驶距离成本通过各路径段距离  $d_{ij}$  乘以  $C_1$  求和获得；

#### 2. 时间窗成本：

软时间窗机制的设计是为了适应生鲜配送场景中“时效优先”与“成本控制”的双重诉求。以新天地超市为例，生鲜货架通常在早间客流高峰期（8:00-10:00）进行补货；配送车辆过早到达（早于 7:30）时超市人员尚未完成货架整理，将导致卸货延迟；晚于 12:00 到达则错失补货黄金时段，影响客户购物体验。所以时间窗惩罚成本（ $C_3$ ）被赋予较高权重（40 元/分钟），显著高于等待成本（ $C_2=10$  元/分钟）。这样的差异化定价策略引导算法在求解时优先规避迟到风险：通过调整发车顺序或增加临时车辆；从而在成本可控范围内最大化准时送达率。故设计如下：

当车辆提前到达节点  $i$ （即  $t_i^k < e_i$ ）时，产生等待成本  $C_2 (e_i - t_i^k)$

当车辆迟到（即  $t_i^k > l_i$ ）时，产生惩罚，成本为  $C_3 (t_i^k - l_i)$

$$P_i = \begin{cases} C_2(e_i - t_i^k), & \text{早于最早服务时间} \\ C_3(t_i^k - l_i) & \text{晚于最早服务时间} (i, j \in n, k \in m) \\ 0, & \text{其他} \end{cases} \quad (5-1)$$

因此，目标函数可表示为：

$$\min Z = \sum_{k=1}^K [C_0 + C_1 \sum_{(i,j) \in R_k} d_{ij} + \sum_{i \in R_k} (C_2 * \max\{e_i - t_i^k, 0\} + C_3 * \max\{t_i^k - l_i, 0\})] \quad (5-2)$$

其中  $R_k$  为车辆  $k$  服务的节点集合， $d_{ij}$  通过平面坐标计算得到。

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5-3)$$

### 5.1.3 模型约束

为确保配送方案的合理性，模型需满足下列约束：

1. 车辆容量约束公式为：

$$\sum_{i \in R_k} q_i \leq Q, \quad \forall k = 1, 2, \dots, K \quad (5-4)$$

用以确保每辆车所配送货物总量不超过其载重上限。

2. 时间窗约束公式为：

对于车辆  $k$  在服务连续节点  $i$  和  $j$  之间，需满足：

$$t_i^k \geq t_i^k + s_i + \frac{d_{ij}}{v} - M(1 - x_{ij}), \quad \forall i, j, k \quad (5-5)$$

其中  $M$  为足够大的常数，用于保证非相邻节点之间的松弛条件。

3. 客户服务约束公式为：

每个客户节点必须恰好被一辆车访问一次：

$$\sum_{k=1}^K \sum_{j=0}^n x_{ij} = 1, \quad \forall i = 1, \dots, n \quad (5-6)$$

4. 路线连通性（配送中心）约束公式为：

每辆车必须从配送中心出发，并在完成配送后返回配送中心：

$$\sum_{j=1}^n x_{0j} = \sum_{i=1}^n x_{i0} = K \quad (5-7)$$

5. 客户点流量平衡约束公式为：

$$\sum_{k=1}^K x_{ijk} = \sum_{i=1}^N x_{ijk} \quad (k \in m, \forall j = 1, 2, \dots, N) \quad (5-8)$$

用以确保进出车辆数相等；

6. 重量约束：每辆车的载重量不能超过其最大载重限制，公式为

$$\sum_{i=1}^N \sum_{j=0}^N x_{ijk} m_j \leq M(k \in m) \quad (5-9)$$

7. 客户点服务约束：确保每个客户点被服务 1 次

$$\sum_{k=1}^K \sum_{i=0}^N x_{ijk} = (j = 1, 2, \dots, N) \quad (5-10)$$

8. 车辆行驶距离约束：使每辆车配送不超过最大配送距离

$$\sum_{i=0}^N \sum_{j=0}^N x_{ijk} d_{ij} \leq D(k \in m) \quad (5-11)$$

## 5.2 带软时间窗的 VRP 模型建立

在前述符号定义和目标函数的基础上，本节详细构建混合整数规划模型，将车辆路径问题与软时间窗约束结合，并说明数据来源和模型求解的具体步骤。

模型构建过程中既需要考虑行驶成本，又通过等待与惩罚项反映配送时效对服务质量的

影响。模型的核心目标函数为：

$$\min Z = \sum_{k=1}^K [C_0 + C_1 \sum_{(i,j) \in R_k} d_{ij} + \sum_{i \in R_k} (C_2 * \max\{e_i - t_i^k, 0\} + C_3 * \max\{t_i^k - l_i, 0\})] \quad (5-12)$$

本模型通过以下步骤构建：

1. 目标函数构建：将车辆启动、行驶与时间窗成本集成在一起，形成总成本函数。
2. 约束条件设置：包括车辆容量、时间窗、客户服务和路线连通性等多重约束，确保模型反映实际配送需求。
3. 参数确定：

距离  $d_{ij}$  由节点平面坐标计算得出；成本系数  $C_0, C_1, C_2, C_3$  依据企业运营数据和问卷调查结果确定。例如，经层次分析法(AHP)计算，“准时性”权重约为 37.02%，故对迟到的惩罚成本  $C_3$  设为较高值(如 40)，以确保在求解过程中尽量降低迟到风险。数据预处理与参数设定结合实验报告，数据处理流程为：

1. 站点数据获取：利用百度地图及坐标拾取器获得南岗区 14 个新天地超市的经纬度，并在 Excel 中整理，再通过 Python 脚本进行数据校验，确保每个数据点的准确性。
2. 坐标转换：通过 Python 程序将经纬度转换为 XY 坐标，确保距离计算准确；具体转换公式为：

$$x = R * \Delta\lambda * \cos(center_{lat}), y = R * \Delta\phi \quad (5-13)$$

其中  $\Delta\lambda$  与  $\Delta\phi$  分别为经纬度的弧度差。

模型求解思路为：构建好的数学模型交由混合整数规划求解器处理，考虑到 VRPTW 问题的 NP 难性，本研究引入改进的粒子群算法(PSO)作为启发式求解方法，通过在目标函数中引入较大惩罚项，对不满足约束的解进行修复，最终实现全局最优或近似最优解的搜索。

### 5.3 设计粒子群算法

粒子群算法模仿鸟群觅食行为，每个粒子代表一种可能的配送路径方案。其更新公式：

$$\begin{aligned} v_i(t+1) &= w * v_i(t) + c_1 * r_1 * (pBest_i - x_i(t)) + c_2 * r_2 * (gBest_i - x_i(t)) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \quad (5-14)$$

其中,  $w$  为惯性因子,  $c_1, c_2$  分别为自我认知和社会认知系数,  $r_1, r_2$  为随机数。通过不断迭代更新, 粒子趋于全局最优解。

这里采用的编码策略是将每个粒子用一排列序列表示客户节点的访问顺序, 解码时采用贪婪算法将序列划分为多条路线, 确保各路线满足车辆容量与时间窗要求。构建适应度函数时综合考虑: 行驶距离成本 (各路径段距离乘以  $c_1$ )、车辆启动成本 ( $c_0$  乘以车辆数) 和时间窗成本 (等待与迟到产生的费用) 三者。

对于不满足约束条件的粒子则通过加入较大惩罚项使其适应度值显著提高, 使气在群体中不易被选为最优解。

鉴于带软时间窗的 VRP 模型求解复杂度较高, 本文采用的改进的粒子群算法 (PSO) 为提升算法性能, 在传统 PSO 基础上进行下面的改进:

(1) 自适应惯性权重: 初始阶段设置较高  $w=0.9$ , 鼓励粒子广泛探索解空间, 避免遗漏潜在优质路径; 随着迭代次数增加线性降低  $w$  至 0.4, 令粒子逐步聚焦于当前最优解邻域, 提升局部寻优精度。

(2) 局部搜索算子: 针对生鲜配送中常见的“集群配送”特征 (如多个超市集中于某商业街), 采用“节点交换”与“路径插入”策略——在每次迭代中对当前最优解进行邻域调整 (交换、插入操作): 若相邻粒子路径中存在地理位置邻近的客户节点, 则交换其服务顺序以减少绕行距离; 若某车辆剩余载重允许, 则将孤立节点插入现有路线, 降低车辆启动频次以进一步降低整体成本。

(3) 交叉操作: 结合顺序交叉策略使粒子间信息共享, 保证群体多样性, 加速收敛过程。

---

## 5.4 配送路径优化模型求解

本节详细描述如何利用 Python 对带软时间窗的 VRP 模型进行求解, 包括数据预处理、初始解生成、粒子群迭代更新以及结果展示。首先进行数据预处理与坐标转换。

### 1. 站点数据采集与校验:

在百度地图中利用搜索功能确定南岗区 14 个新天地超市的具体位置, 并通过坐标拾取器获取经纬度。随后, 将数据录入 Excel 表格, 通过比对实际地图标注, 确保数据准确性。

### 2. 经纬度转换为平面坐标:

采用 Python 代码进行转换, 转换结果用于构建各节点间的欧式距离矩阵, 保证后

续适应度计算的准确性。

之后,进行初始解生成与车辆分配。初始解构造为利用贪婪策略生成初始配送序列,确保每个序列能在短时间内构造出一条可行解;贪婪算法依据节点之间距离依次选择最近未访问的节点,直到构成一条完整路线,然后判断该路线是否满足车辆载重和时间窗要求,若不满足则启动新车辆配送;

车辆分配与适应度计算则使用 `calFitness` 函数对每个配送序列进行解码,将序列划分成若干条可行路线,同时计算每条路线的总行驶距离、等待时间及迟到惩罚。最后,将各项成本相加,得到每个粒子的综合适应度值。

粒子群迭代更新方面设定最大迭代次数(如 1000 次),不断更新粒子位置,通过 `crossover` 函数实现交叉操作,并结合局部搜索算子对当前最优解进行微调。每次迭代后记录全局最优适应度,并根据预设的收敛条件判断是否终止迭代。

结果可视化则利用 `draw_path` 函数通过 `matplotlib` 绘制配送路径图,直观展示各车辆的具体配送路线。

## 5.5 实例分析

本节以哈尔滨新天地超市南岗区配送案例为例,详细分析模型求解过程及结果,验证所提模型和算法在实际场景中的有效性。

### 5.5.1 实证背景概述

实验区域为南岗区,共涉及 14 个新天地超市配送点。背景说明如下:

1. 在百度地图中使用搜索功能确定南岗区 14 个新天地超市位置和配送点位置并标记。



图 5-2 新天地超市位置和配送点位置

表 5-1 新天地超市位置和配送点位置坐标

	lat	lon
1	45.75013	126.6423
2	45.74604	126.699
3	45.75437	126.7111
4	45.74492	126.6791
5	45.76604	126.6815
6	45.76939	126.6923
7	45.76224	126.6541
8	45.76321	126.6633
9	45.77163	126.6589
10	45.75548	126.6502
11	45.75223	126.6448
12	45.76082	126.6397
13	45.76928	126.6311
14	45.72825	126.604
配送点	45.77699	126.8139
中心点	45.72	126.6

在获取经纬度信息后，进行了数据校验工作。通过对比百度地图上的实际位置和标注的经纬度信息，确保数据的准确性。然后利用 Python 将经纬度转换 XY 坐标代码，以（45.72, 126.60）为南岗区中心坐标，运行代码后得出经纬度转换 XY 坐标结果。对于存在偏差的数据点，重新查询和修正。

表 5-2 经纬度转换 XY 坐标信息

	lat	lon	X/km	Y/km
1	45.75013	126.6423	3.286677	3.354056
2	45.74604	126.699	7.678368	2.898203
3	45.75437	126.7111	8.635016	3.826162



4	45.74492	126.6791	6.143798	2.773525
5	45.76604	126.6815	6.336153	5.125149
6	45.76939	126.6923	7.174435	5.498292
7	45.76224	126.6541	4.20602	4.702469
8	45.76321	126.6633	4.918473	4.810449
9	45.77163	126.6589	4.575575	5.747537
10	45.75548	126.6502	3.90338	3.949616
11	45.75223	126.6448	3.483151	3.587605
12	45.76082	126.6397	3.084917	4.54395
13	45.76928	126.6311	2.417074	5.485825
14	45.72825	126.604	0.309012	0.918052
配送点	45.77699	126.8139	16.62566	6.343541
中心点	45.72	126.6	0	0

配送中心选址固定设于道外区团结镇郭地方路 66-68 号传化公路港 B 区，具有较好的物流配套设施与交通网络优势。这里本文使用节点空间分布图直观显示客户和配送中心在区域内的分布情况，了解地理位置对配送路径规划的影响。

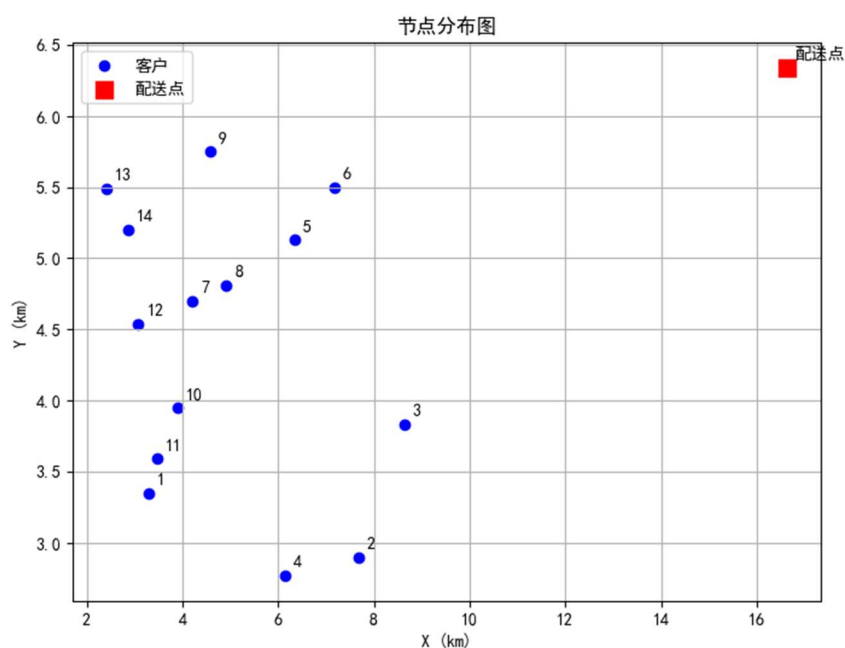


图 5-3 配送点位置空间分布图

接下来的需求预测与时间窗设定基于周边人口密度、商业辐射范围及历史销售数据，

对各超市的日均需求量进行预测；同时，结合各超市营业时间、交通状况、配送成本及客户高峰期设定合理的配送时间窗（例如 8:00~12:00），确保生鲜产品配送的时效性和品质。图 5.4 与 5.5 分别是各节点需求量的柱状图以及需求量和 service 时长关系散点图，展示各节点之间在服务负荷上的差异并探讨需求量与服务时长之间的相关性、反映客户间的异质性，为后续的车辆负载分配和资源优化提供依据。

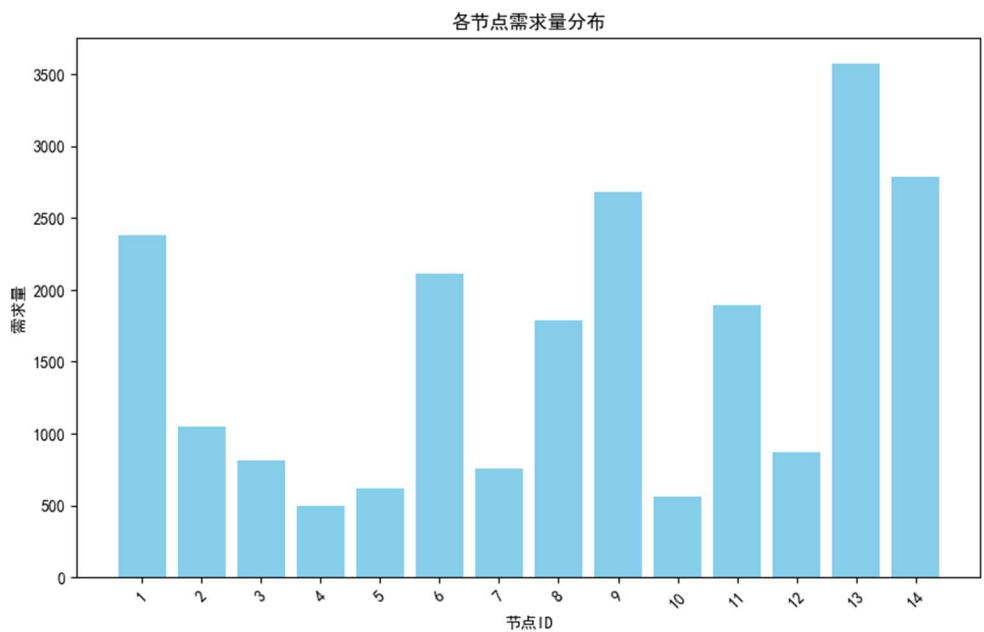


图 5-4 各节点需求量分布柱状图

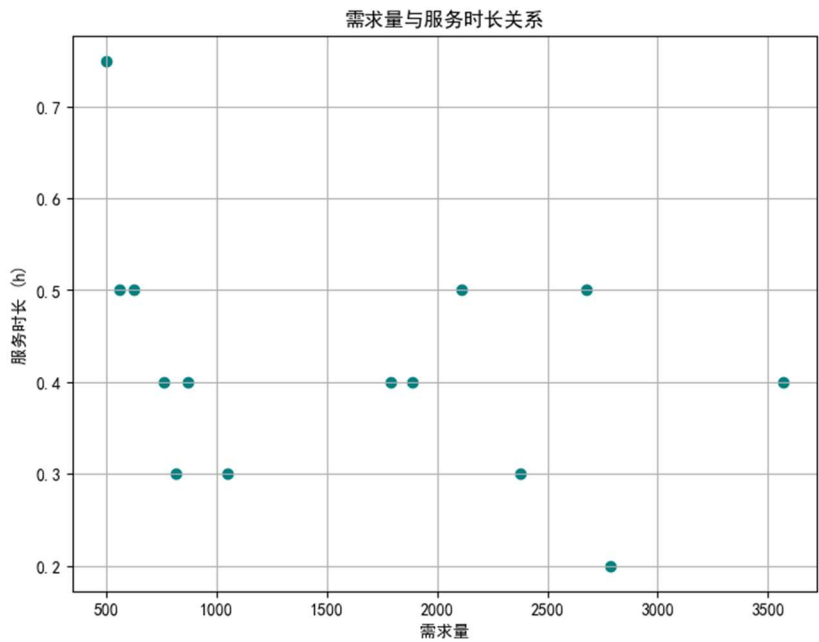


图 5-5 需求量与服务时长关系散点图

这里的需求量和人数柱状图很容易看出各客户之间存在明显的差异，如节点 1、9、

13、14 的需求量明显远大于其它节点；而服务时长分布图及需求量与服务时长的散点图反映了不同客户在服务需求上的差异，其中的关系需要使用数学方法来捕捉以得出最优方案，进一步验证了配送过程中使用本文的类似粒子群优化（PSO）的思想来不断改进解考虑客户间异质性的现实情况是非常有必要的，为在模型中设置不同的配送策略提供了数据支持。

2. 通过调查问卷收集了客户对配送服务的评价，主要指标包括准时性、商品丢失、配送员服务态度、费用和包装质量。利用层次分析法得出影响配送服务的最大权重

(1) 将两指标重要性比值赋值（重要为 3、比较重要 2、相同重要为 1），得出以下判断矩阵：

表 5-3 判断矩阵

G(A)	准时性	商品丢失	配送员服务态度	配 送 费 用	包 装 质 量
准时性	1	3	3	2	2
商品丢失	0.33	1	2	2	2
配送员服务态度	0.33	0.5	1	0.5	0.5
配送费用	0.5	0.5	2	1	1
包装质量	0.5	0.5	2	1	1

通过将判断矩阵按行相乘，计算  $w_i$  的平均值得：

$w_i \ (i = 1,2,3,4,5) = 2.047672511, 1.214285434, 0.528548434, 0.870550563, 0.870550563$

计算判断矩阵的特征向量得：

$W = (0.370176754, 0.219517642, 0.095550603, 0.157377501, 0.157377501)^T$

以及计算得最大特征根为 5.15969751；

计算一致性指标得：

$C.I. = 0.039924379$

计算一致性比率得：

$C.R. = 0.035646767 < 0.1$

说明判断矩阵合理，调查结果显示，准时性得分最低。

经层次分析法（AHP）计算，准时性在各项指标中的权重约为 37.02%，提示在优化模型时应重点关注时间窗约束。

3. 根据各站点所处的地理位置、人口数据以及消费水平等因素，对各站点的需求量进行详细预测。考虑客户需求、配送中心运营时间、交通状况和路况、配送成本以及节假日和特殊事件等多个因素根据各站点流量高峰期，设置时间窗据开始送货的

开始时间、结束时间和服务时长，得下表 5.4。

表 5-4 新天地超市需求列表

节点 ID	X/km	Y/km	人数	需求量	时间窗开始	时间窗结束	EET/h	LLT/h	服务时长
1	3.29	3.35	334	2378	6:30	8:00	1	2.5	0.3
2	7.68	2.90	212	1048	8:30	10:00	3	4.5	0.3
3	8.64	3.83	189	817	9:30	11:00	4	5.5	0.3
4	6.14	2.77	153	501	13:00	17:00	7.5	11.5	0.75
5	6.34	5.13	167	623	10:00	13:00	4.5	7.5	0.5
6	7.17	5.50	257	2110	10:00	13:00	4.5	7.5	0.5
7	4.21	4.70	168	761	12:10	14:20	6.67	8.83	0.4
8	4.92	4.81	214	1790	6:45	8:00	1.25	2.5	0.4
9	4.58	5.75	370	2678	9:15	11:00	3.75	5.5	0.5
10	3.90	3.95	178	561	11:45	13:10	6.25	7.67	0.5
11	3.48	3.59	259	1890	14:30	16:20	9	10.83	0.4
12	3.08	4.54	167	871	15:00	17:00	9.5	11.5	0.4
13	2.42	5.49	558	3570	9:30	11:30	4	6	0.4
14	2.87	5.20	427	2784	8:40	10:00	3.17	4.5	0.2
配送点	16.63	6.34							

### 5.5.2 模型求解

在 Python 环境中，依据前述参数设置和数据预处理，模型求解过程为：

1. 参数设置：

车辆参数： $Q = 5000$ ；最大行驶距离 $D = 1000$ ；车辆速度 $V = 40 \text{ km/h}$ 。

成本系数： $C_0 = 100$ ， $C_1 = 2$ ， $C_2 = 10$ ， $C_3 = 40$ 。

PSO 参数：粒子数 50，迭代次数 1000，惯性因子 $w=0.2$ ，学习因子 $c_1=c_2=0.4$ 。

2. 初始解构造与适应度计算：

利用贪婪算法生成初始配送序列，通过 calFitness 函数进行车辆分配和时间窗检测，计算出每个解的总成本（适应度值），并确定当前全局最优解。

3. 迭代更新:

在预设迭代次数内不断更新粒子位置,利用交叉操作和局部搜索进一步改善解的质量。实验表明,由图 5-7 可知,算法在约第 135 次迭代时收敛,得到的最优配送路径能有效降低车辆总行驶距离和时间窗违约成本。

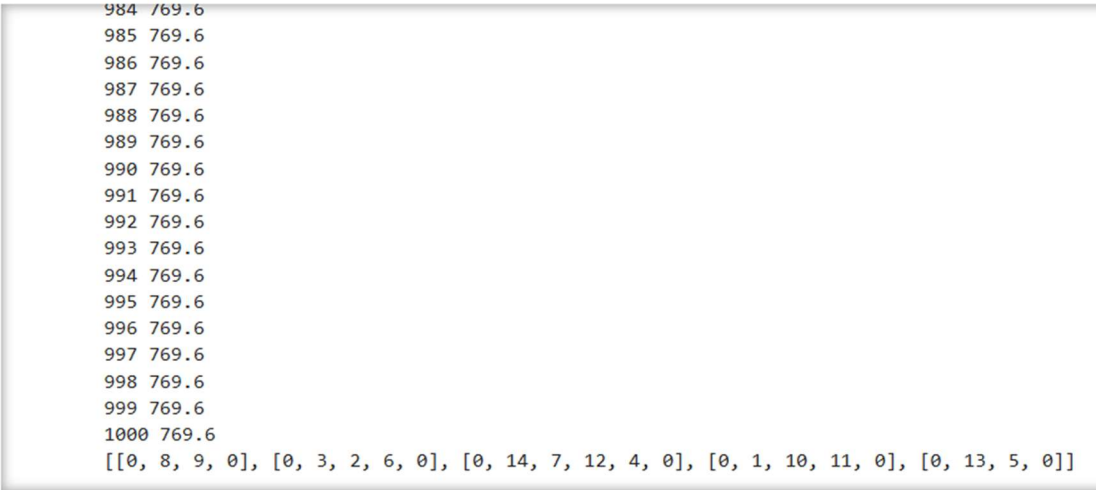


图 5-6 运行带时间窗的 VRP 模型代码后得到的迭代结果 1

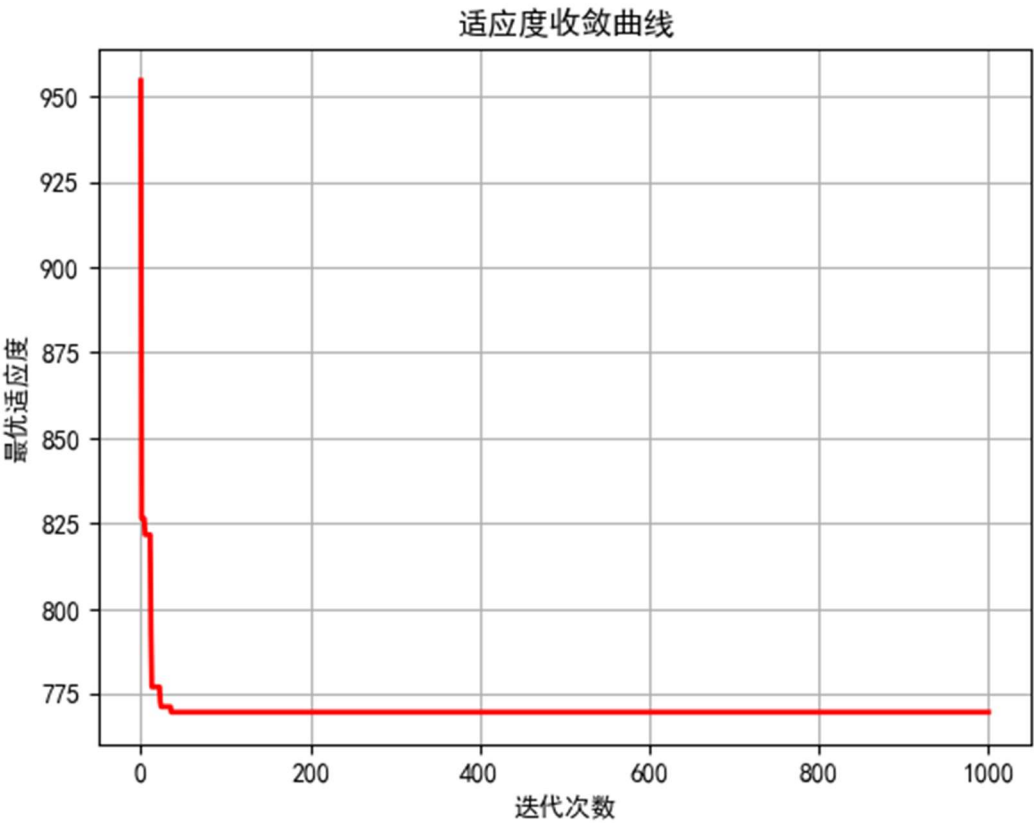


图 5-7 运行带时间窗的 VRP 模型代码的适应度收敛曲线

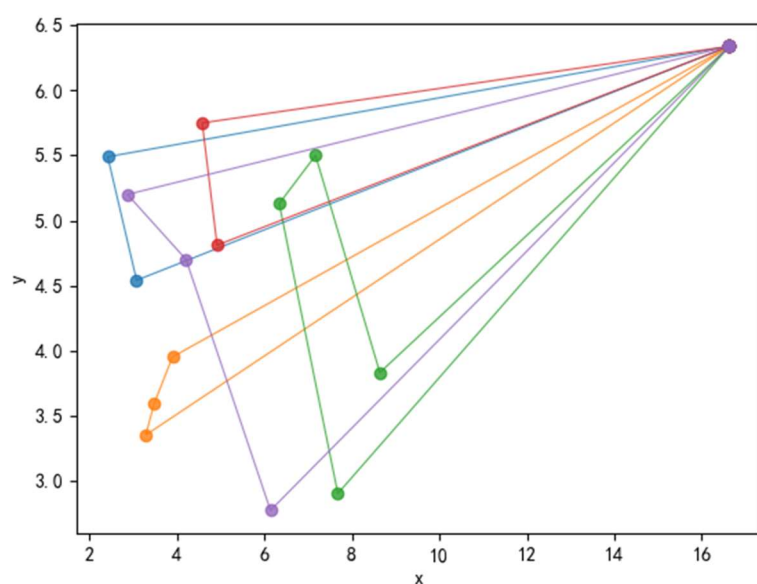


图 5-8 运行带时间窗的 VRP 模型代码后得到的迭代结果 2

### 5.5.3 计算结果与讨论

“适应度”是目标函数由车辆行驶距离、车辆启动车数、等待成本和迟到惩罚成本按一定权重加权得到的。适应度值越低，说明配送方案在总成本上越优秀。在实验中发现，直接采用贪婪算法构造的初始解（未进行迭代时），其适应度（即总配送成本）约为 826.3；而采用改进粒子群算法经过 1000 次迭代后，最优解的适应度降至 769.6。也就是说，优化后的方案相比初始方案成本降低了约 6.8%。

1	826.3
2	826.3
3	826.3
4	826.3
5	821.6
6	821.6
7	821.6
8	821.6
9	821.6
10	821.6
11	821.6
12	792.6
13	777.0
14	777.0
15	777.0
16	777.0
17	777.0
18	777.0

图 5-9 未进行迭代时的贪婪算法结果

