Modeling and Simulation 建模与仿真, 2025, 14(4), 238-252
Published Online April 2025 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2025.144282

Hans汉斯

考虑洪水淹没的应急物资调运方案优化

王欣妍，黄中意

上海理工大学管理学院，上海

收稿日期：2025年3月10日；录用日期：2025年4月3日；发布日期：2025年4月14日

摘　要

本文研究了洪灾背景下考虑洪水淹没的应急物资调运优化问题，目标是在车辆和道路受限的情况下，将低海拔地区的应急物资快速转移到高海拔仓库。论文提出了两种求解方法：遗传算法和一种分阶段的混合算法(近似解法)，并通过一个25个物资储备库的算例进行了对比分析，得出混合算法在求解效率和结果优越性方面更具优势的结论。

关键词

洪水淹没，应急物资调运，遗传算法，近似解法，组合优化

Optimization of Emergency Material Dispatch Scheme Considering Flood Submergence

Xinyan Wang, Zhongyi Huang

Business School, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 10th, 2025; accepted: Apr. 3rd, 2025; published: Apr. 14th, 2025

Abstract

This paper investigates the optimization problem of emergency material transportation considering flood inundation in the context of flood disasters. The objective is to quickly transfer emergency supplies from low-altitude areas to high-altitude warehouses under the constraints of limited vehicles and road access. The paper proposes two solution methods: a genetic algorithm and a phased hybrid algorithm (an approximate solution method). Through a comparative analysis using a case study of 25 material reserve depots, it concludes that the hybrid algorithm has advantages in both solution efficiency and result superiority.

文章引用: 王欣妍, 黄中意. 考虑洪水淹没的应急物资调运方案优化[J]. 建模与仿真, 2025, 14(4): 238-252.
DOI: 10.12677/mos.2025.144282

Keywords

Flood Inundation, Emergency Material Transportation, Genetic Algorithm, Approximate Method, Combinatorial Optimization

1. 引言

近年来，由于全球气候变暖导致极端天气事件频发，如暴雨、台风等，随着水位的不断上涨会引起洪涝灾害发生，一旦发生洪灾，低海拔地区的应急物资储备库就容易被淹没，从而不但阻碍了接下来的救援工作，而且影响之后对受灾群众的物资供给。中国是洪涝灾害特别严重的国家，这些事件往往导致严重的人员受难和物资损失，和对应急物资需求的急速增加。在洪灾来临前，将现有应急物资转运至相对安全地区以防止物资受损，进而更好地开展接下来的救援工作。其中，由于低海拔地区容易积水导致物资储备库被淹，所以可以将物资转运至高海拔安全地区以减少应急物资受损。因此，探究洪灾下考虑洪水淹没的应急物资转运是一个重要的课题。

应急物资调运问题是一个典型的组合优化问题，涉及车辆路径规划、任务分配和资源调度等多个方面的问题，由于其 NP 难特性，近年来，启发式算法[1]-[6]逐渐成为解决应急物资调运问题的主流方法，这些算法通过模拟自然进化或群体行为，能够在较短时间内找到近似最优解，但其随机性较强，容易陷入局部最优解。

逐渐地，混合算法的出现逐渐弥补了启发式算法的不足，徐超毅和刘晓絮[7]为减少洪涝灾害时的经济损失，构建考虑灾区对应急物资的需求程度约束和软时间窗约束的路径优化模型，考虑车辆行驶速度受天气差异影响的变化，设计混合遗传算法求解；黄晨煜[8]结合当前交通路网的路段受损率来计算配送区域内路径的广义运输距离，建立应急物流车辆配送路径优化模型，针对模型特点提出了考虑“先分区，后规划”思想的二阶段混合算法来完成求解算法设计；钟媛媛[9]构建了综合多环境因素影响的带软时间窗的应急物资配送车辆路径问题模型，并设计了一种混合蚁群禁忌搜索算法进行求解；金瑜杰[10]将应急物资调运问题分成物资分配和车辆路径规划两部分，并分别用遗传算法和蚁群算法求解；Zhang 等[11]提出了一种结合人工免疫算法和蚁群算法的混合算法来求解应急车辆路径问题，在较短时间内得到了更优的解；Wu 等[12]提出了一种基于蚁群算法和变邻域搜索算法相结合的混合算法，并将其与常用优化算法进行比较，验证了该混合算法可以在较短时间内求得问题的解。

本文针对现有研究提出一种混合算法，并将应急物资调运问题分成运输问题、任务分配问题和车辆路径规划问题三部分，建立不同的模型并用不同的方法求解。

2. 考虑洪水淹没的应急物资调运问题

2.1. 问题描述

在车辆有限，道路通行受限和洪水即将来临的背景下，以最短的时间内将低海拔点的应急物资全部运输至高海拔点仓库为总目标，规划应急物资调运方案，以防止应急物资被即将来临的洪水淹没和受损。该问题的描述及相关假设如下：

1) 物资情况：有 m 个低海拔物资储备点 $A_1, A_2, \cdots, A_m$，各低海拔点的现有物资量分别为 $a_1, a_2, \cdots, a_m$。

有 n 个高海拔物资储备点 $B_1,B_2,\cdots,B_n$，各高海拔点的物资容量分别为 $b_1,b_2,\cdots,b_n$。a_i' 为低海拔点的剩余物资量，b_i' 为高海拔点的储存物资的剩余空间。高海拔点的物资总容量大于低海拔点的物资总量，即 $\sum a_i \le \sum b_i$。

2) 路网：任意两点 $P_i\left(x_i,y_i\right)$ 与 $P_j\left(x_j,y_j\right)$（P_i 和 P_j 可是任意低海拔点或是高海拔点)间的运输距离为 d_{ij}，距离使用欧几里得公式求解：

$$d_{ij}=\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2} \tag{1}$$

3) 运输车辆：有 K 辆卡车($K\le m$)，每辆车的最大容量为 Q，卡车平均速度为 V。

4) 运输规则：卡车 k 可去任意地点装卸货，每次装卸货的时间为固定值 T；卡车 k 形成一个由装卸货任务构成的运输方案 s_k：

$$s_k=\left\{s_{k,1},\cdots,s_{k,i},\cdots,s_{k,n}\right\}=\left\{\left(P_1,L_{k,1}\right),\cdots,\left(P_i,L_i\right),\cdots,\left(P_n,L_n\right)\right\} \tag{2}$$

上式中 $s_{k,i}$ 表示第 i 次装卸货任务，$\left(P_i,L_i\right)$ 即为本次装卸货任务的物资点以及具体的装卸量。当 $L_i>0$ 时表示在点 P_i 装货，当 $L_i<0$ 时表示在点 P_i 卸货。当完成一次装卸货任务时，需该物资储备点的货物量更新。当该点是低海拔点时，更新物资量 a_i：

$$a_i=a_i-L_i \tag{3}$$

当该海拔点是高海拔点时，更新物资容量 b_i：

$$b_i=b_i+L_i \tag{4}$$

卡车 k 在运输全程中任意时刻的容量需小于等于最大容量 q。即对于任一次装卸货任务 $s_{k,j}$，执行完该任务后卡车总容量应不超过 q：

$$\forall s_{k,j},\sum_{i=1}^{j}L_i\le q \tag{5}$$

当 s_k 执行完成时，卡车应当清空：

$$\sum_{i=1}^{n}L_i=0 \tag{6}$$

根据运输方案 s_k，可求得卡车 k 的运输总时长 t_k 为：

$$t_k=\sum_{i=1}^{n+1}\frac{d_{i,i+1}}{V}+nT \tag{7}$$

5) 初始情况与结束条件：初始情况为卡车 k 在空车时，同时在某低海拔点开始执行 s_k。完成条件为所有低海拔点的物资全部运输完成且所有卡车完成运输任务：

$$\sum a_i=0 \text{ and } \sum_{i=1}^{n}L_i=0 \tag{8}$$

6) 目标函数：所有卡车的总运输时间最小：

$$\min z=\max\left(\left\{t_k:k=1,2,\cdots,k\right\}\right) \tag{9}$$

2.2. 问题分析

该问题是一个典型的组合优化问题。当我们关注其中一辆车的运输方案 s_k 时，可近似看作是由一系

列车辆路径问题(Vehicle Routing Problem, VRP)组成的集合。

VRP 的基本假设如下：

1) 假设有一个中心仓库(起点)，以及若干个客户点，每个客户点有固定的需求量。

2) 设计一组车辆的行驶路径，使得每辆车从仓库出发，经过若干个客户点，再返回仓库。

3) 每辆车的行驶路线需要满足容量限制(即每辆车的最大载货量不能超过其容量)，并且路径应当尽量优化(通常是最小化总距离、总时间或总成本)。

VRP 的约束条件如下：

1) 容量约束：每辆车的运输能力有限，每次只能运输一定数量的货物。

2) 路径约束：每个客户点必须由一辆车访问，且每个客户点仅能被访问一次。

s_k 中的运输方案与 VRP 的主要区别在于每次完成卸货之后，车辆无需返回中心仓库，而是去往下一个仓库，且并无“每个客户点必须由一辆车访问，且每个客户点仅能被访问一次”的约束，因此每辆车的运输方案比 VRP 问题中的单车运输方案具有更大的组合空间，可将此问题归约为一个 VRP 问题。由于 VRP 问题是一个 NP 难问题，因此考虑洪水淹没的应急物资调运问题本质上也是一个 NP 难问题，不可在多项式时间内求解该问题，通常可使用启发式算法，如贪心算法、遗传算法、蚁群算法等进行求解。

3. 遗传算法求解

1) 概述

本研究采用遗传算法(Genetic Algorithm, GA)来求解物资运输调度问题，旨在找到一种最优的车辆运输方案，使得所有低海拔物资储备点的物资在最短时间内被清空并运输至高海拔物资储备点。遗传算法通过模拟自然选择和遗传机制，在解空间中搜索最优解。

2) 编码方式

每个个体(运输方案)由 K 辆车的运输路线组成，其中每辆车的运输路线被编码为一个长度为 L 的整数序列，序列中的每个整数表示物资储备点的编号，范围从 0 到 $m+n-1$，其中 0 到 $m-1$ 代表低海拔点，m 到 $m+n-1$ 代表高海拔点。例如，对于第 k 辆车，其运输路线为 $\{p_1, p_2, \cdots, p_L\}$，其中 p_i 车辆 k 在第 i 步访问的物资储备点编号。

3) 初始种群的生成

通过随机生成 r 个个体来构建。对于每个个体，每辆车的运输路线通过随机选择 L 个物资储备点编号来生成，如公式(10)所示：

$$\text{population} = \left\{ \left(\left[\text{randint}\left(0, m+n-1\right) \right]_{i=1}^{L} \right)_{k=1}^{K} \right\}_{j=1}^{r} \tag{10}$$

其中，population 是初始种群，j 表示种群中的个体编号，k 表示车辆编号，i 表示车辆运输路线中的步骤编号。

4) 适应度函数设计

适应度函数用于评估每个个体(运输方案)的优劣。根据适应度对个体进行排序，记录当前代的最优适应度和最优个体。

在本问题中，适应度定义为所有低海拔物资储备点清空时的时间，若在模拟过程中低海拔物资储备点未清空，则适应度为无穷大(∞)。

具体计算过程如下：

对于每个个体(即 K 辆车的运输方案)，初始化低海拔点的现有物资量 a、高海拔点的物资容量 b、每辆车的当前任务索引 ti、当前负载 tl 和下一更新时间点 t'，以及全局时间 t。

在模拟过程中，每辆车按照其运输路线依次访问物资储备点。当车辆到达低海拔物资储备点时，它会装载尽可能多的物资，直到车辆满载或该低海拔点物资清空。装载量 load 由公式(11)确定：

$$\text{load} = \min\left(Q - tl[k], a[\text{loc}]\right) \tag{11}$$

其中，$a[\text{loc}]$ 是当前低海拔点的现有物资量，loc 是是当前访问低海拔点编号。

若 load > 0，则更新低海拔点现有物资量 $a[\text{loc}]$ 和车辆载货量 $tl[k]$，并增加装货时间 T，如公式(12)、(13)、(14)所示：

$$a[\text{loc}] = a[\text{loc}] - \text{load} \tag{12}$$

$$tl[k] = tl[k] + \text{load} \tag{13}$$

$$T = T + t' \tag{14}$$

当车辆到达高海拔物资储备点时，它会卸载尽可能多的物资，直到车辆空载或该高海拔点物资容量已满。卸载量 unload 由公式(15)确定：

$$\text{unload} = \min\left(tl[k], b[\text{loc} - m]\right) \tag{15}$$

若 unload > 0，则更新高海拔点物资容量 $b[\text{loc} - m]$ 和车辆载货量 $tl[k]$，并增加卸货时间 T，如公式(16)、(17)所示：

$$b[\text{loc} - m] = b[\text{loc} - m] - \text{unload} \tag{16}$$

$$tl[k] = tl[k] - \text{unload} \tag{17}$$

每次车辆移动到下一个物资储备点时，计算移动时间 t''，移动时间 t'' 由公式(18)确定：

$$t'' = \frac{d[\text{loc}][\text{next}_{\text{loc}}]}{V} \tag{18}$$

其中，$d[\text{loc}][\text{next\_loc}]$ 是当前位置 loc 到下一个位置 next_loc 的距离。

更新下一更新时间点 $t'[k]$ 和当前任务索引 $ti[k]$，如公式(19)、(20)所示：

$$t'[k] = t'[k] + t'' \tag{19}$$

$$ti[k] = ti[k] + 1 \tag{20}$$

重复上述过程，直到所有低海拔物资储备点清空或确认无法清空。若所有低海拔物资储备点清空，则适应度为当前全局时间 t；若未清空，则适应度为∞。适应度函数 fitness 可表示为公式(21)：

$$\text{fitness} = \begin{cases} t, & \sum_{i=0}^{m-1} a_i = 0 \\ \infty, & \sum_{i=0}^{m-1} a_i \neq 0 \end{cases} \tag{21}$$

5) 选择算子

采用基于适应度排序的选择策略，选择适应度较好的个体进入下一代种群，以保留优良基因。

6) 交叉算子

使用单点交叉操作，在每对父代个体(每辆车的运输路线)中随机选择一个交叉点，交换交叉点之后的部分，生成两个子代个体。

7) 变异算子

对每个个体(每辆车的运输路线)以一定的变异概率进行变异操作。变异时随机选择运输路线中的一个位置，将其对应的物资储备点编号替换为一个随机生成的编号。

当算法的迭代次数大于预先设置的迭代数值时，便输出最优解，结束运算。具体结构流程如图 1 所示。

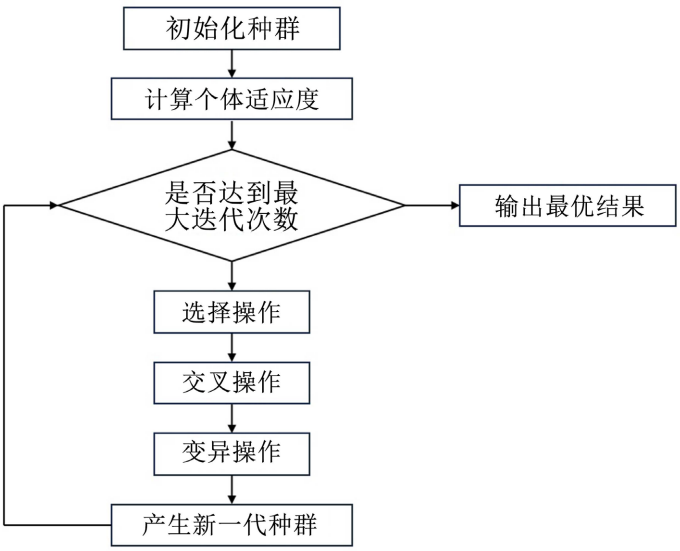

Figure 1. Flowchart of genetic algorithm
图 1. 遗传算法流程图

4. 模型构建及求解方法

本文从车辆使用的角度出发，提出一种考虑洪水淹没的应急物资调运问题的近似解法。该解法的总体思路可描述为：让所有车辆尽可能高效运行，不同车辆的运输工作量尽可能相同，且优先保证大部分物资(如 80%以上)的运输任务。为实现上述思路，将该问题拆分为依次求解的 3 个子问题：

1) 运输问题：为保证车辆高效运行，应使货物的总体运输的时间成本最小。可先不考虑车辆运输过程中的约束，将该问题看作由 m 个产地(低海拔点)和 n 个销地(高海拔点)的运输问题，使得整体运输方案的距离成本最小，完成低海拔点和高海拔点的匹配。

2) 任务分配问题：为保证不同车辆的运输工作量尽可能相同，将低海拔点聚成 K 簇，使得簇内点间的距离尽可能小、簇间的距离尽可能大，且不同簇对应的总运输量尽可能相同。每辆车负责一个簇的运输任务，完成车辆任务的分配。

3) 车辆路线规划问题：为优先保证大部分物资的运输任务，车辆将按照低海拔点物资量的顺序，从大到小、由近及远的依次完成簇内运输任务。

接下来将分别介绍各子问题的数学模型及其求解方法。

4.1. 运输问题模型

1) 决策变量

x_{ij} 为从低海拔点 i 到高海拔点 j 的运输量； y_{ij} 为低海拔点 i 是否运输至高海拔点 j，是为 1，否则为 0。

2) 模型建立

$$\min z = \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} x_{ij} \tag{22}$$

$$\sum_{j=1}^{n} x_{ij} = a_i,\ i \in m \tag{23}$$

$$\sum_{i=1}^{m} x_{ij} \le b_j,\ j \in n \tag{24}$$

$$\sum_{j=1}^{n} y_{ij} \le \left\lceil \frac{Q}{a_i} \right\rceil,\ i \in m \tag{25}$$

$$x_{ij} \le M y_{ij},\ i \in m,\ j \in n \tag{26}$$

$$x_{ij} \ge 0,\ y_{ij} \in (0,1),\ i \in m,\ j \in n \tag{27}$$

目标函数(22)为运输总路径最短；约束条件(23)为运输量等于低海拔点的现有物资总量；约束条件(24)为运输量不大于高海拔点的最大物资总容量；约束条件(25)为低海拔点 A_i 向高海拔点 B_j 运输的节点数不能超过 $\frac{Q}{a_i}$ 的上取整函数，以减少车辆的运行路线；约束条件(26)为运输量与决策变量的关系，M 是一个足够大的常数，用于确保当 $y_{ij}=0$ 时， $x_{ij}=0$；约束条件(27)为非负约束。

3) 模型求解

运用表上作业法求解该应急物资分配模型，算法如下：

Step 1：列出平衡表。基于 d_{ij} 得到一个 $m \times n$ 的距离表，并将距离表填入最终的平衡表。将配送区域内路网的各路段以“未受损可通行”和“完全受损不可通行”两种情况，若某节点路段为完全受损中断通车状态，在程序运行时将其距离设定为一个无限大的正数 M，如表 1 所示。

Table 1. Balance sheet
表 1. 平衡表

	B_1	B_2	…	B_n	a_i
A_1	d_{11}	d_{12}	…	d_{1n}	a_1
A_2	d_{21}	d_{22}	…	d_{2n}	a_2
…	…	…	$\boldsymbol{M}$	…	…
A_m	d_{m1}	d_{m2}	…	d_{mn}	a_m
b_j	b_1	b_2	…	b_n	

Step 2：运用伏格尔法确定运输方案，得到应急物资分配结果表，如表 2 所示。由于该模型是基于应急物资分配问题改进的运输模型，且接下来还要进行优化，因此得到基本初始可行解即可。另外，如果 $x_{ij}=0$，可以不在表格上体现。

Table 2. Allocation table of emergency material
表 2. 应急物资分配表

	B_1	B_2	…	B_n	a_i'
A_1	x_{11}	x_{12}	…	x_{1n}	a_1'
A_2	x_{21}	x_{22}	…	x_{2n}	a_2'

续表

…	…	…	…	…	…
A_m	x_{m1}	x_{m2}	…	x_{mn}	a'_m
b_j	b'_1	b'_2	…	b'_n	

4.2. 任务分配问题模型

本阶段需要将总区域的 m 个低海拔点分为 K 类：对 m 个低海拔点使用改进的 K-means 算法将所有的低海拔点划分至 K 个子区域，并将与低海拔点有运输量的高海拔点划分至该低海拔点所属的子区域中。

1) 问题定义

给定的数据集 $A=\{A_i \mid i=1,2,\cdots,m\}$，给定的数据的容量为 a_i，目标数据块数目为 K；平衡容量约束聚类将根据预先定义的距离将目标样本集 A 平衡地划分为 K 簇，每个簇都具有一个聚类中心点，共有 K 个聚类中心点，记为 $B=\{\mu_k \mid k=1,2,\cdots,K\}$；聚类结果中所有的簇信息存储在 C_k 中，C'_K 表示簇的容量；e_{ik} 为指示变量，若节点 i 被分配到 C_k，则为 1，否则为 0；l_{ik} 表示节点 i 到簇中心点 μ_k 的距离；s_{min} 表示每个簇至少包含的数据点个数；β 表示容量平衡惩罚系数，用于平衡距离成本和容量偏差。

2) 模型建立

本问题在传统 K-means 算法的基础上，提出了一种基于容量平衡的 K-means 聚类算法，其目标函数由两部分组成：节点到簇中心点的距离成本以及簇容量与目标容量之间的偏差惩罚。每个簇的总容量尽可能接近平均容量。对于每个簇的 C_K，计算其总容量与平均容量的差值的绝对值，进而对所有簇的这些绝对值求和。

$$F=\min\sum_{k=1}^{K}\left(\sum_{i\in C_K} l_{ik}+\beta\cdot\left|C'_K-\frac{\sum_{i=1}^{m}a_i}{K}\right|\right) \tag{28}$$

$$\left|C_k\right|\geq s_{\min},\ k=1,2,\cdots,K \tag{29}$$

$$\sum_{k=1}^{K}e_{ik}=1 \tag{30}$$

目标函数(28)为使每个聚类的大小尽可能接近平均值，避免某些聚类过大或过小；约束条件(29)为每个簇至少包含的数据点个数；约束条件(30)为每个节点只能分配到一个簇。

3) 优化步骤

Step 1：初始化中心点。采用 K-means++方法初始化簇的中心点，并确保初始中心尽可能分散，随机初始化 K 个聚类中心 μ_k。

Step 2：节点分配。对于每个节点，计算其到所有簇中心点的距离，并结合容量惩罚项，选择使总成本最小的簇进行分配。分配点到最近的聚类中心。根据点与聚类中心的加权距离分配仓库到各聚类。

$$d_i=\sqrt{\left(x_i-\mu_{kx}\right)^2-\left(y_i-\mu_{ky}\right)^2} \tag{31}$$

Step 3：中心点更新。对于每个簇，重新计算其中心点，使其为簇内节点的加权距离最小点。聚类中心取属于该簇的点的加权平均。

$$\mu_k=\frac{\sum_{i\in C_k}w_i\left(x_i,y_i\right)}{\sum_{i\in C_k}w_i} \tag{32}$$

Step 4：调整簇的容量。若当前容量小于最佳容量，更新最佳簇和最佳容量，确保每个簇容量接近目标容量。

Step 5：迭代优化。重复上述 Step 2 和 Step 3，直至达到最大迭代次数或目标函数收敛，输出最优解，结束运算。具体结构流程如图 2 所示。

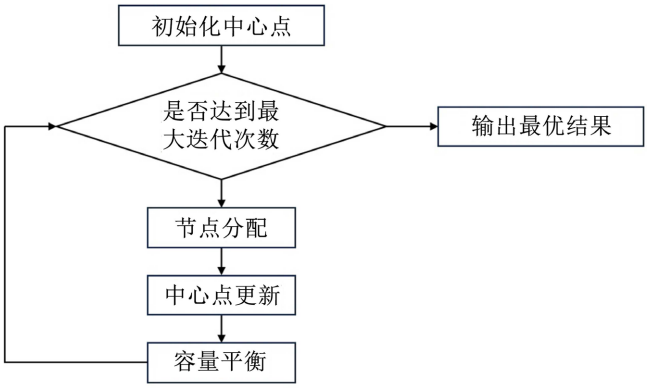

Figure 2. Clustering algorithm flowchart
图 2. 聚类算法流程图

4.3. 车辆路径规划问题模型

根据 3.1 和 3.2，得到卡车 k 的运输任务集合 T_k 如下：

$$T_k = \left\{ \left(A_i, B_j, x_{ij} \right) \middle| A_i \in C_k \right\} \tag{33}$$

为优先保证大部分货物的运输任务，卡车 k 按照如下贪心算法运行：

Step 1：抵达运输量 x_{ij} 最大的低海拔点 A_i，转入 Step 3。

Step 2：找到所有运输量 $x_{ij} \geq Q$ 的运输任务，若为空集，转入 Step 1；若不为空集，则抵达离当前节点最近的低海拔节点 A_i，转入 Step 3。

Step 3：若 $x_{ij} \geq Q$，转入 Step 4；若 $x_{ij} < Q$，转入 Step 5。

Step 4：执行任务 $\left\{ \left(A_i, +Q \right), \left(B_j, -Q \right) \right\}$，$x_{ij} = x_{ij} - Q$，若 $x_{ij} < Q$，即无法装满一辆车，转入 Step 5；若 $x_{ij} \geq Q$，转入 Step 4。

Step 5：若 $x_{ij} \geq \lambda Q$，则转入 6；若 $x_{ij} < \lambda Q$，将 $\left(A_i, B_j, x_{ij} \right)$ 从任务中删除，若删除后任务列表为空，则结束，否则转入 Step 2。

Step 6：执行任务 $\left\{ \left(A_i, +x_{ij} \right), \left(B_j, -x_{ij} \right) \right\}$，将 $\left(A_i, B_j, x_{ij} \right)$ 从任务中删除，若删除后任务列表为空，则结束，否则转入 Step 2。

以上算法中，λ 为物资储备点剩余物资系数，当剩余物资量超过 λQ 时，车辆选择将该低海拔点的物资清空，否则将不处理该点的剩余物资，转而重新进入 Step 1 完成运输量最大的任务。经过 Step 1 之后，每个低海拔点的物资剩余量将不超过 λQ。

5. 算例分析

5.1. 数据准备

本文选取某市 25 个物资储备库进行算例分析，通过查找相关统计资料、新闻资讯以及专家评估等方

式获取各个物资储备库的数据，如表 3 和表 4 所示。对于个别无法获取的数据，结合已有数据和实际情况进行虚拟设置，不影响模型的有效性和实际意义，本文设定车辆最大容量 Q 为 30 吨，车辆运行速度 V 为 50 km/h，物资点剩余储备物资系数 λ 为 0.1，装卸货时间 T 为 0.3 h。

Table 3. Specific data for low altitude points
表 3. 低海拔点具体数据

A_i	坐标	a_i
A_1	(−29.730, 64.136)	54
A_2	(−30.664, 5.463)	38
A_3	(51.642, 5.469)	40
A_4	(−13.171, 69.336)	55
A_5	(−67.413, 68.323)	54
A_6	(48.907, 6.274)	41
A_7	(5.243, 22.260)	34
A_8	(−65.002, 77.234)	52
A_9	(−4.175, −1.569)	39
A_{10}	(23.029, 11.639)	49
A_{11}	(25.482, 6.287)	55
A_{12}	(−42.615, −26.392)	65
A_{13}	(−76.672, 99.341)	56
A_{14}	(−20.673, 57.892)	47
A_{15}	(−52.039, 6.567)	52
A_{16}	(−41.376, 50.824)	38

Table 4. Specific data for high altitude points
表 4. 高海拔点具体数据

B_j	坐标	b_i
B_1	(−37.756, −33.325)	99
B_2	(23.767, 29.083)	138
B_3	(−43.030, 20.453)	89
B_4	(−35.297, −24.896)	93
B_5	(−54.755, 14.368)	88
B_6	(−49.329, 33.374)	79
B_7	(57.404, 23.822)	92
B_8	(−22.754, 55.408)	83
B_9	(−56.622, 73.340)	108

基于表 3 和表 4 节点的坐标信息，可绘制运输区域内的坐标分布图，如图 3 所示。

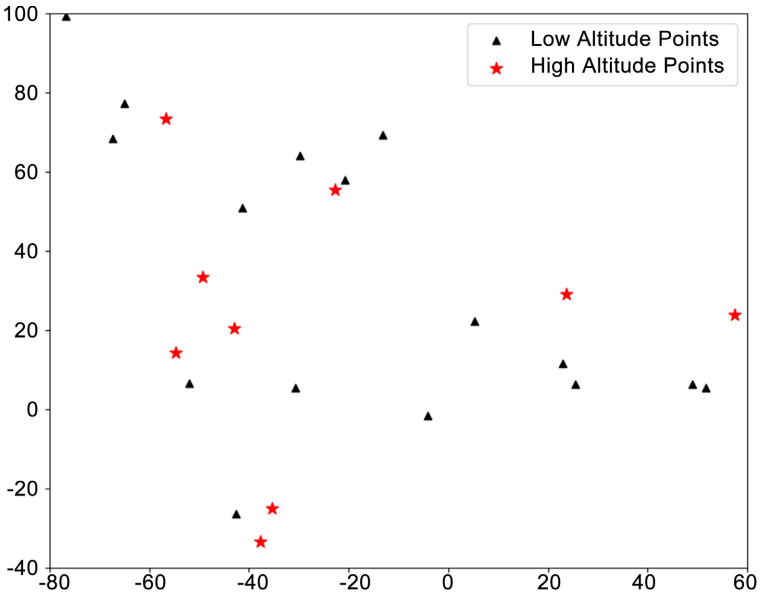

Figure 3. Location distribution map of all coordinate points within the transportation area
图 3. 运输区域内所有坐标点的位置分布图

5.2. 分阶段求解

1) 运输问题

本文主要通过随机生成的方式来对路段受损情况做假设，确定某节点路段为完全受损中断通车状态，并在程序运行时将其距离设定为一个无限大的正数 M，由此即完成了对路段通行信息数据的调整得到平衡表，如表 5 所示。

Table 5. Balance sheet
表 5. 平衡表

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	a_i
A_1	97	63	45	89	55	36	96	***M***	28	54
A_2	***M***	59	19	30	25	33	89	50	72	38
A_3	97	36	***M***	92	106	***M***	19	89	127	40
A_4	105	54	57	96	68	50	83	16	43	55
A_5	***M***	99	53	98	55	39	132	46	11	54
A_6	95	33	93	***M***	***M***	101	19	86	125	41
A_7	70	19	48	62	60	55	52	43	80	34
A_8	113	100	60	106	63	46	***M***	47	9	52
A_9	***M***	41	44	38	53	57	66	59	91	39
A_{10}	75	17	66	68	77	75	***M***	63	***M***	49
A_{11}	74	22	69	68	80	79	36	68	106	55
A_{12}	8	86	***M***	7	42	60	111	84	100	65

续表

A_{13}	138	122	85	***M***	87	71	153	69	32	56
A_{14}	92	52	43	84	55	37	***M***	3	39	47
A_{15}	42	***M***	16	35	8	26	110	56	66	52
A_{16}	84	68	30	75	38	19	102	19	27	38
b_j	99	138	89	93	88	79	92	83	108	

基于得到的平衡表和运输模型得到的结果为 20,911，具体的应急物资分配方案如表 6 所示。

Table 6. Plan table of emergency material distribution
表 6. 应急物资分配方案表

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	a_i'
A_1			24		30					0
A_2			16	22						0
A_3							40			0
A_4			25					30		0
A_5			24			30				0
A_6							41			0
A_7		34								0
A_8									52	0
A_9				39						0
A_{10}		49								0
A_{11}		55								0
A_{12}	65									0
A_{13}									56	0
A_{14}								47		0
A_{15}					52					0
A_{16}						38				0
b_i'	34	0	0	32	6	11	11	6	0	

2) 任务分配问题

基于 2.2 介绍的改进 *K*-means 聚类算法，利用 Python 软件实现对低海拔点的聚类分析，设迭代 1000 次，容量惩罚函数设为 30，容量期望值为 202.75，每个簇至少包含的数据点个数 $s_{\min}$ 为 4。得到的结果如表 7 所示。展示聚类划分结果如图 4 所示。

3) 车辆规划问题

通过表 7 可知，每一簇有一条应急物资运输路线，根据车辆路径规划问题模型，得到每辆车的最优行驶时间，通过三阶段的分段求解得到的总行驶时间的最优值为 40.09，每辆车的具体运输方案如表 8 所示。

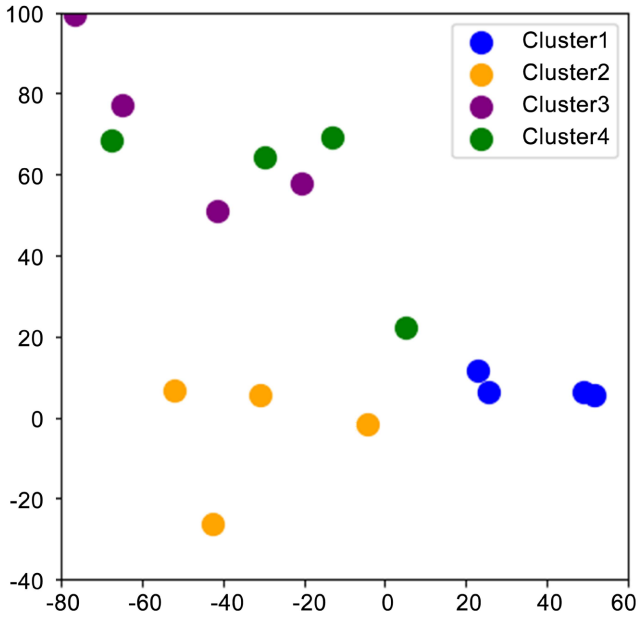

Figure 4. Cluster division result chart
图 4. 聚类划分结果图

Table 7. Result table of cluster division
表 7. 聚类划分结果表

C_k	A_i	B_j	总容量	与期望值的差
C_1	A_3、A_6、A_{10}、A_{11}	B_2、B_7	208	5.25
C_2	A_2、A_9、A_{12}、A_{15}	B_1、B_3、B_4、B_5	206	3.25
C_3	A_8、A_{13}、A_{14}、A_{16}	B_5、B_6、B_8、B_9	189	13.75
C_4	A_1、A_4、A_5、A_7	B_2、B_3、B_5、B_6、B_8	208	5.25

Table 8. Plan table of transportation
表 8. 运输方案表

C_k	s_k	t_k
C_1	('A10', 30), ('B2', −30), ('A10', 19), ('B2', −19), ('A11', 30), ('B2', −30), ('A11', 25), ('B2', −25), ('A6', 30), ('B7', −30), ('A6', 11), ('B7', −11), ('A3', 30), ('B7', −30), ('A3', 10), ('B7', −10)	8.72
C_2	('A9', 30), ('B4', −30), ('A9', 9), ('B4', −9), ('A12', 30), ('B1', −30), ('A12', 30), ('B1', −30), ('A12', 5), ('B1', −5), ('A15', 30), ('B5', −30), ('A15', 22), ('B5', −22), ('A2', 16), ('B3', −16), ('A2', 22), ('B4', −22)	9.49
C_3	('A14', 30), ('B8', −30), ('A14', 17), ('B8', −17), ('A16', 30), ('B6', −30), ('A16', 8), ('B6', −8), ('A8', 30), ('B9', −30), ('A8', 22), ('B9', −22), ('A13', 30), ('B9', −30), ('A13', 26), ('B9', −26)	7.56
C_4	('A7', 30), ('B2', −30), ('A7', 4), ('B2', −4), ('A4', 25), ('B3', −25), ('A4', 30), ('B8', −30), ('A1', 24), ('B3', −24), ('A1', 30), ('B5', −30), ('A5', 24), ('B3', −24), ('A5', 30), ('B6', −30)	14.32

5.3. 遗传算法求解

交叉概率为 0.9，变异概率为 0.1，初始种群大小为 200，个体任务序列长度为 200，迭代次数设为 50。使用 Python 进行运算，得到目标函数最优值为 64.4，具体的遗传算法运输方案如表 9 所示。

Table 9. Plan table of genetic algorithm transportation
表 9. 遗传算法运输方案表

C_k	s_k
C_1	('B2', 0), ('A5', 30), ('B2', −30), ('B0', 0), ('A12', 30), ('A15', 0), ('A11', 0), ('A1', 0), ('A7', 0), ('A12', 0), ('B1', −30), ('A13', 0), ('B3', 0), ('A11', 30), ('B5', −30), ('A11', 30), ('A0', 0), ('A1', 0), ('A0', 0), ('A2', 0), ('B3', −30), ('B7', 0), ('A0', 24), ('B8', −24), ('A4', 24), ('B8', −24), ('B2', 0), ('A15', 0), ('A9', 0), ('A8', 26), ('B4', −26), ('A12', 8), ('B7', −8), ('A1', 8), ('A6', 4), ('B5', −12), ('A10', 0)
C_2	('B5', 0), ('A13', 17), ('A8', 13), ('A6', 0), ('B1', −30), ('B5', 0), ('A9', 19), ('A13', 0), ('B0', −19), ('B0', 0), ('B7', 0), ('A3', 30), ('A4', 0), ('A2', 0), ('A6', 0), ('A8', 0), ('B4', −30), ('A11', 5), ('A2', 25), ('A2', 0), ('B2', −30), ('A6', 30), ('A6', 0), ('B0', −30), ('A14', 0), ('A11', 0), ('A2', 15), ('B7', −15), ('A10', 25), ('A12', 5), ('A13', 0), ('B4', −30), ('A15', 0), ('A5', 0), ('A11', 0), ('A9', 0), ('A6', 0), ('A0', 0), ('A13', 0)
C_3	('A13', 30), ('A13', 0), ('A2', 0), ('A5', 0), ('A8', 0), ('A11', 0), ('A9', 0), ('B1', −30), ('A1', 30), ('A7', 0), ('B5', −30), ('A13', 0), ('A10', 30), ('B1', −30), ('A15', 30), ('A12', 0), ('A0', 0), ('B6', −30), ('A7', 30), ('A9', 0), ('B0', −30), ('B6', 0), ('A15', 8), ('A2', 0), ('A4', 0), ('A11', 0), ('B3', −8), ('A0', 0), ('A3', 17), ('A4', 0), ('A4', 0), ('A12', 13), ('B6', −30), ('A13', 0), ('A11', 0)
C_4	('A9', 30), ('A4', 0), ('A12', 0), ('A4', 0), ('B8', −30), ('A13', 0), ('B8', 0), ('B4', 0), ('A5', 11), ('B3', −11), ('B7', 0), ('A4', 30), ('A8', 0), ('B3', −30), ('B4', 0), ('A14', 30), ('B7', −30), ('A14', 22), ('B7', −22), ('A9', 0), ('B3', 0), ('A0', 30), ('B6', −30), ('A2', 0), ('A7', 22), ('A3', 8), ('A6', 0), ('A4', 0), ('A9', 0), ('A12', 0), ('A15', 0), ('B6', −2), ('B2', −28)

5.4. 对比分析

1) 运行原理对比

遗传算法：在本文的应急物资调运问题中，遗传算法通过随机生成初始种群，并通过适应度函数评估每个个体的优劣，逐步优化运输方案。然而，遗传算法的随机性较强，容易陷入局部最优解，且计算量较大，尤其是在处理大规模问题时，收敛速度较慢。

混合算法：混合算法从车辆使用的角度出发，将问题分解为运输问题、任务分配问题和车辆路径规划问题三个子问题，依次求解。通过分阶段优化，混合算法能够有效降低问题的复杂度，减少计算量，避免遗传算法的随机性，提高求解的稳定性。

2) 稳定性对比

遗传算法：遗传算法的随机性较强，尤其是在初始种群生成和变异操作中，容易受到随机因素的影响，导致每次运行的结果可能存在较大差异。这种不稳定性在实际应用中可能导致求解结果不可靠，尤其是在应急物资调运这种对时间敏感的场景中，遗传算法的表现可能不够稳定。

混合算法：混合算法通过分阶段求解，每个子问题的求解过程都具有较高的确定性。运输问题模型和任务分配问题模型均基于确定性算法(如表上作业法和改进的 *K*-means 聚类算法)，能够保证每次运行的结果一致。车辆路径规划问题虽然使用了贪心算法，但其基于前两个阶段的确定性结果，整体求解过程仍具有较高稳定性。

3) 计算量对比

遗传算法：遗传算法的计算量较大，尤其是在种群规模较大、迭代次数较多的情况下，每次迭代都需要计算每个个体的适应度函数，并进行交叉和变异操作。对于本文的应急物资调运问题，遗传算法需要处理多个车辆的运输路线组合，计算复杂度较高，导致求解时间较长。

混合算法：混合算法通过分阶段求解，将复杂的问题分解为多个简单的子问题，每个子问题的求解过程相对独立，计算量较小。运输问题模型和任务分配问题模型均可以通过高效的算法(如表上作业法和 *K*-means 聚类算法)快速求解，车辆路径规划问题也通过贪心算法在较短时间内得到可行解。因此，混合

算法的计算量显著低于遗传算法，如表 10 所示。

Table 10. Different operating schedules for two algorithms
表 10. 两种算法的不同运行时间表

算法类型	混合算法	遗传算法
运行时间(秒)	0.04	422

4) 结果对比

混合算法的最优行驶时间为 40.09 小时，而遗传算法的最优行驶时间为 64.4 小时。混合算法的行驶时间显著优于遗传算法，表明混合算法能够更高效地完成应急物资调运任务。

5) 总结

通过对比分析可以看出，混合算法在稳定性、计算量和求解效率等方面均优于遗传算法。混合算法通过分阶段求解，能够有效降低问题的复杂度，减少计算量，并在较短时间内得到较优解。此外，混合算法的求解过程具有较高的确定性，适合应急物资调运这种对时间敏感的场景。因此，混合算法为洪灾应急物资调运方案优化提供了更为有效的方法。

参考文献

[1] 杨莎莎. 考虑道路可靠性的地震灾害应急物资调度优化[D]: [硕士学位论文]. 徐州: 中国矿业大学, 2022.

[2] 张忆. 基于蚁群算法的城市应急救援路径规划研究[D]: [硕士学位论文]. 成都: 西南财经大学, 2023.

[3] 周丹, 邱玉琢. 基于改进蚁群优化算法求解应急物资库存路径问题[J]. 粮食科技与经济, 2021, 46(6): 76-79.

[4] 郑尔宗, 张以晨, 徐毓蔓. 基于改进遗传算法的地震应急物资配送路径规划[J]. 华北地震科学, 2024, 42(2): 25-29+72.

[5] 徐创宇. 疫情下考虑需求紧迫度的应急医疗物资配送路径优化研究[D]: [硕士学位论文]. 南昌: 江西财经大学, 2023.

[6] 马骏博. 基于改进粒子群算法灾后应急物资运送车辆调度优化研究[D]: [硕士学位论文]. 沈阳: 沈阳大学, 2024.

[7] 徐超毅, 刘晓絮. 洪涝灾害下考虑需求紧迫度的应急物资配送研究——以安徽省为例[J]. 南阳理工学院学报, 2023, 15(6): 12-21.

[8] 黄晨煜. 道路受损条件下应急物流配送路径优化与决策研究[D]: [硕士学位论文]. 长沙: 中南林业科技大学, 2022.

[9] 钟媛媛. 改进蚁群优化算法在灾后应急物资配送路径规划中的研究[D]: [硕士学位论文]. 咸阳: 西北农林科技大学, 2024.

[10] 金瑜杰. 突发事件下应急物资优化配置建模研究[D]: [硕士学位论文]. 上海: 东华大学, 2022.

[11] Zhang, Q. and Xiong, S. (2018) Routing Optimization of Emergency Grain Distribution Vehicles Using the Immune Ant Colony Optimization Algorithm. *Applied Soft Computing*, **71**, 917-925. https://doi.org/10.1016/j.asoc.2018.07.050

[12] Wu, Y., Pan, F., Li, S., Chen, Z. and Dong, M. (2019) Peer-Induced Fairness Capacitated Vehicle Routing Scheduling Using a Hybrid Optimization ACO-VNS Algorithm. *Soft Computing*, **24**, 2201-2213. https://doi.org/10.1007/s00500-019-04053-9