

卡车 - 无人机联合物流的建模与优化研究

姚钊艺

上海理工大学管理学院, 上海

收稿日期: 2025年3月25日; 录用日期: 2025年4月18日; 发布日期: 2025年4月27日

摘要

建立单卡车与多无人机协同配送且无人机可携带多需求的物流网络, 基于最小化配送的总成本, 构建无人机多次访问且与卡车协同配送的线性混合整数规划模型, 使用自适应大邻域搜索算法对其进行求解, 最后通过求解算例论证模型和算法的效果。研究表明, 无人机 - 卡车协同配送比单独卡车配送节约了更多的成本, 而携带多无人机的卡车 - 无人机配送系统相较于仅携带单无人机的卡车 - 无人机系统和单独卡车配送的系统表现更好, 自适应大邻域搜索算法求解此模型也更优于CPLEX求解器。

关键词

卡车 - 无人机车辆路径问题, 混合线性整数规划, 自适应大邻域搜索

Modeling and Optimization Study of Truck UAV Joint Logistics

Zhaoyi Yao

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

Received: Mar. 25th, 2025; accepted: Apr. 18th, 2025; published: Apr. 27th, 2025

Abstract

Establish rural background order truck and UAV collaborative distribution logistics network, based on minimizing the summary cost, build the drones multiple visit and truck with distribution linear mixed integer planning model, using the adaptive large neighborhood search algorithm to solve the effect of the model and algorithm. Research shows that drones-truck collaborative distribution than separate truck distribution saves more cost, and carry more drones truck-drone distribution system compared with only a single drone truck-drone system and separate truck distribution system is better, adaptive large neighborhood search algorithm solving this model is better than the CPLEX solver.

Keywords

Truck-Drone Vehicle Path Issue, Mixed-Linear Integer Programming, Adaptive Large-Neighborhood Search

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来,随着所有零售公司迅速从实体店转移到电子商务渠道,零售业的销售方式发生了根本的转变,可以说互联网是将传统零售业务转变为电子商务的原因,而同期,在大数据、人工智能的驱动下,线上线下市场双向联动,新零售或将代替电商,成为全新的商业模式[1]。而在线零售的加速发展,快递配送量的持续增长,物流服务商面临“最后一公里”配送的诸多挑战[2]。据中国国家统计局(2021a)通报的数据,2020年社会消费品零售总额39.198万亿元人民币,其中在线零售的实物产品销售额9.759万亿元人民币。2019年年底新冠肺炎疫情爆发,受此影响,2020年社会消费品零售总额同比下降3.9%,然而在线零售额和快递配送量逆势增长14.5%和31.2%,近五年的年均增长率分别为24.8%和32.5%,在线零售业与快递服务业市场规模持续扩大[3]。因此,研究如何使用无人机的配送方式解决最后一公里问题有重要的现实意义。

已有国内外学者对带有无人机协助的车辆路径问题进行研究。Murray等(2015)对于加入无人机串联配送旅行商问题(FSTSP)和无人机并联配送的旅行商问题(PDSTSP)进行了研究。这两类问题的出现用于解决仅含有一组客户和一个仓库且一个客户只被卡车或者无人机服务一次,这些研究均以配送时间最小化建立目标。在无人机卡车并联配送模式下,大多数情况针对一组客户点距离仓库均不是很远,此时无人机不跟随卡车同步配送,仅需要从仓库出发,完成配送后回到仓库,循环往复,卡车则单独出发配送相对更远的客户;而在无人机卡车串联配送模式下,卡车会携带无人机一起进行配送,此模式下无人机在某一卡车到达的节点会被释放去完成配送,且单次释放携带单一需求,卡车在此过程则继续前往下一个客户点,并在发射无人机之后的某一顾客点回收无人机[4]。针对上述两种问题,后续的学者们展开了一系列的拓展。Schermer等(2019)题出的FSTSP就是最初的一种无人机-卡车配送思想,用于“最后一公里交付”,并对单卡车与多无人机以最小化返程时间进行了建模研究,并允许无人机携带多需求[5]。Dayarian等(2020)提出带有无人机支持的配送新逻辑模型,该模型下卡车等待所有发射的无人机作业完成后出发前往下一客户点,卡车于此情形下更多作为辅助去支持无人机完成配送[6]。最后多卡车与多无人机配送问题也得到相关研究,如Wang等(2019)对多辆卡车与多无人机共同提供配送服务,但此类问题的复杂程度常常难以解决,本篇不再过多叙述。国内也有部分研究黄凯明等(2018)在带有无人机辅助的旅行商问题的基础上研究车载无人机三层协同路径规划问题[7]。刘想等(2019)以物资配送为场景,研究了单卡车多无人机协同配送的双层路径规划问题,采用基于节约思想、回归思想的启发式算法求解[8]。

在算法优化方面,刘想等(2019)考虑时变性的单卡车与单无人机联合配送的路径规划问题,建立MILP模型并采用分支切割算法和变邻域搜索算法求解[8]。Boccia等(2019)以最小化行驶成本和客户等待时间为优化目标构建携带无人机的旅行商问题模型,采用改进蚁群算法和K-means迭代聚类算法求解[9]。

综上所述,现有文献对卡车-无人机协同配送及对其进行算法优化方面进行了深入研究,卡车-无人机联合配送的灵活性决定其并无固定模型,单卡车单无人机配送效率受限制,而多卡车多无人机问题

会十分复杂导致在大规模问题中难以解决,因此单卡车多无人机在串联工作背景下可以进行更多的优化,先前的研究也确实缺乏部分的讨论,比如对于模拟实际农村环境算例以及对卡车-无人机系统配送整体性能优化。此类问题求解使用了各种不同的启发式算法,效果各不相同,而求解质量应同样作为研究的一环[10]。因此,本文对于求解加入卡车-无人机作业路线的考虑,针对单卡车与多无人机,无人机可携带多需求且与卡车协同的配送系统建模,最后使用自适应大邻域搜索算法进行较大规模随机生成算例优化求解,提升整体的求解质量。

2. 联合配送建模与算法优化分析

2.1. 假设与参数

- 1) 每架无人机可携带多需求
 - 2) 无人机可以在起始点、终点及卡车访问的顾客点发射或回收
 - 3) 卡车在能够装载所有无人机的同时可以携带所有的需求量
 - 4) 发射或回收无人机所产生的时间忽略不计
 - 5) 卡车与无人机服务顾客点的服务时间忽略不计
- N, s, d : 顾客点集合, s 为起始点, t 为终点;
 N_s : $N \cup s$, 顾客点与起点的集合;
 N_t : $N \cup t$, 顾客点与终点的集合;
 N_{st} : $N \cup s \cup t$, 顾客点与起点和终点的集合;
 A : 所有路线的集合 ($A = \{(i, j) | i \in N_s, j \in N_t, i \neq j\}$);
 D : 所有无人机 d 的集合;
 d_{ij}^K : 卡车从 i 点到 j 点的距离;
 d_{ij}^D : 无人机从 i 点到 j 点的距离;
 t_{ij}^K : 卡车从 i 点到 j 的旅行时间;
 t_{ij}^D : 无人机从 i 点到 j 的旅行时间;
 e : 无人机最大续航时间;
 L : 任意车辆的最大工作时间;
 M : 一个很大的正数;
 x_{ij} : 二进制决策变量, 若卡车经过 (i, j) 为 1, 否则为 0;
 y_{ijd} : 二进制决策变量, 若无人机 d 经过 (i, j) 为 1, 否则为 0;
 z_{ijd} : 二进制决策变量, 若卡车搭载无人机 d 经过 (i, j) 为 1, 否则为 0;
 w_{id} : 二进制决策变量, 若卡车在 i 点发射或回收了无人机 d 为 1, 否则为 0;
 a_{ik}^K : 连续决策变量, 卡车到达 i 点时的时间;
 a_{ik}^D : 连续决策变量, 无人机 d 到达 i 点时的时间;
 b_{id}^- : 连续决策变量, 无人机 d 被发射后到达 i 点累计飞行时间;
 b_{id}^+ : 连续决策变量, 无人机 d 被发射后离开 i 点累计飞行时间。

2.2. 模型建立

$$(p) \min c^K \sum_{(i,j) \in A} d_{ij}^K x_{ij} + c^D \sum_{d \in D} \sum_{(i,j) \in A_s} d_{ij}^D y_{ij} \quad (1)$$

目标函数(1)将总成本最小化

$$s.t. \sum_{j \in N} x_{sj} = 1 \quad (2)$$

$$\sum_{j \in N} x_{it} = 1 \quad (3)$$

约束(2)和(3)则确保卡车必须从起始点 s 离开并最终返回到终止点 t

$$\sum_{i \in N_s} x_{ij} = \sum_{h \in N_t} x_{ih}, \forall j \in N \quad (4)$$

约束(4)为流量守恒, 若卡车在非终点节点到达, 则必须由该节点驶出

$$\sum_{j \in N_t} y_{sjd} + \sum_{j \in N_t} z_{sjd} = \sum_{i \in N_s} y_{itd} + \sum_{j \in N_t} z_{itd}, \forall d \in D \quad (5)$$

约束(5)表示无人机的数量守恒

$$\sum_{i \in N_t} y_{ijd} + \sum_{i \in N_t} z_{ijd} = \sum_{h \in N_t} y_{jhd} + \sum_{h \in N_t} z_{jhd}, \forall j \in N, \forall d \in D \quad (6)$$

约束(6)表示无人机的流量守恒

$$\sum_{i \in N_s} y_{ijd} + \sum_{i \in N_s} z_{ijd} \leq 1, \forall j \in N_t, \forall d \in D \quad (7)$$

约束(7)表示无人机不能重复访问顾客点

$$\sum_{i \in N} x_{ij} \leq 1, \forall j \in N \quad (8)$$

约束(8)表示卡车不能重复访问顾客点

$$z_{ijd} \leq x_{ij}, \forall (i, j) \in A^K, \forall d \in D \quad (9)$$

约束(9)表明约束状态, 卡车搭载无人机时, 该无人机不可独立活动

$$\sum_{i \in N_s} x_{ij} + \sum_{i \in N_s} \sum_{d \in D} y_{ijd} \geq 1 - \sum_{d \in D} w_{jd}, \forall j \in N \quad (10)$$

$$\sum_{i \in N_s} x_{ij} + \sum_{i \in N_s} \sum_{d \in D} y_{ijd} \leq 1 - \sum_{d \in D} w_{jd}, \forall j \in N \quad (11)$$

约束(10)和(11)表示若 j 点不是发射回收节点, 则该点有且仅有卡车或无人机访问一次

$$\sum_{i \in N_s} x_{ij} + \sum_{i \in N_s} y_{ijd} + \sum_{h \in N_t} x_{jh} + \sum_{h \in N_t} y_{jhd} \geq 3w_{jd}, \forall j \in N, \forall d \in D \quad (12)$$

$$\sum_{i \in N_s} x_{ij} + \sum_{i \in N_s} y_{ijd} + \sum_{h \in N_t} x_{jh} + \sum_{h \in N_t} y_{jhd} \leq 2 + 2w_{jd}, \forall j \in N, \forall d \in D \quad (13)$$

约束(12)和(13)共同表示三种特殊路线情形:

j 仅为回收节点, 此时无人机与卡车均访问 j 点, 且之后卡车搭载该无人机离开;

j 仅为发射节点, 此时 j 点仅有卡车访问且在该点发射无人机之后各自前往下一节点;

j 既是发射节点也是回收节点, 此时卡车无人机均访问该点, 且之后再次发射无人机各自离开

$$a_{jk}^K \geq a_{ik}^K + t_{ij}^K - (1 - x_{ij})M, \forall i \in N_s, \forall j \in N_t \quad (14)$$

$$a_{jk}^K \geq a_{id}^D + t_{ij}^D - (1 - x_{ijd})M, \forall i \in N_s, \forall j \in N_t, \forall d \in D \quad (15)$$

约束(14)和(15)构成卡车抵达节点的时间要求

$$a_{jd}^D \geq a_{id}^D + t_{ij}^D - (1 - y_{ijd})M, \forall i \in N_s, \forall j \in N_t, \forall d \in D \quad (16)$$

$$a_{jd}^D \geq a_{ik}^D + t_{ij}^D - (1 - y_{ijd})M, \forall i \in N_s, \forall j \in N_t, \forall d \in D \quad (17)$$

约束(16)和(17)构成无人机抵达节点的时间要求

$$a_{jk}^K \leq L \sum_{i \in N_s} x_{ij}, \forall j \in N_t \quad (18)$$

$$a_{jd}^D \leq L \left(\sum_{i \in N_s} y_{ijd} + \sum_{i \in N_s} z_{ijd} \right), \forall j \in N_t, \forall d \in D \quad (19)$$

约束(18)和(19)构成最大工作时间限制

$$b_{id}^+ \leq (1 - w_{id})M, \forall i \in N_{st}, \forall d \in D \quad (20)$$

约束(20)表示无人机在发射前自身电量已满

$$b_{id}^+ \geq b_{id}^- - Mw_{id}, \forall i \in N, \forall d \in D \quad (21)$$

$$b_{id}^+ \leq b_{id}^- + Mw_{id}, \forall i \in N, \forall d \in D \quad (22)$$

约束(21)和(22)表示在节点配送时忽略耗电

$$b_{jd}^- \geq b_{id}^+ + t_{ij}^D - (1 - y_{ijd})M, \forall i \in N_s, \forall j \in N_t, \forall d \in D \quad (23)$$

约束(23)对无人机抵达顾客点的累计飞行时间进行更新

$$b_{jd}^- \leq e \sum_{i \in N_s} y_{ijd}, \forall j \in N_t, \forall d \in D \quad (24)$$

约束(24)限制无人机的最大续航时间。

2.3. 自适应大邻域算法

自适应大邻域搜索算法(ALNS)是Ropk等在解决基于时间窗的货物交付问题(VRPTW)所提出的一种启发式算法。其基本思想是通过邻域搜索算子的历史表现来赋予其权重, 自适应的破坏和修复, 以达到搜索最优解的目标[11]。算法的核心组成部分包括初始解的构造、算子的设计、算子选择策略和自适应权重调整机制、以及解的接受准则和算法的停止条件等。

通常 ALNS 会有一些破坏算子和修复算子方法作为输入, 先通过 CPLEX 求得一个初始解, 然后根据问题设置相应破坏方法算子和修复方法算子, 通过不断地破坏和修复解来完成对全局的搜索, 得出更优解, 并且会在每一次迭代中更新这些破坏和修复策略的权重, 将权重应用到接下来的破坏以及修复, 不断地对解进行破坏与修复操作, 完成后更新策略方法的权重, 再通过模拟退火验收标准来以一定概率接受现有解, 从而避免陷入局部最优解, 重复此过程直至达到迭代次数上限, 输出最后所求得目标值, 结束循环。

ALNS 算法具体流程表如下表 1 自适应大邻域搜索算法所示:

Table 1. Adaptive large neighborhood search algorithm

表 1. 自适应大邻域搜索算法

Algorithm 1. Adaptive large neighborhood search

- 1: **input:** initial temperature: T_s , iterations: n , maximum iterations without improvement: I_m
- 2: $S_c \leftarrow$ generate initial solution (3.1)
- 3: $S_s \leftarrow S_c$
- 4: $T \leftarrow T_s$
- 5: $I \leftarrow 0$
- 6: **for** iteration $\leftarrow 1$ to n
- 7: $d \leftarrow$ select a destroy operator from φ^- (3.3)
- 8: $r \leftarrow$ select a repair operator from φ^+ (3.3)

续表

```

9:    $S_t \leftarrow \text{apply the operator and } r \text{ to } S_c \text{ (3.2)}$ 
10:  if  $f(S_t) < f(S_b)$  (3.4) then
11:     $S_b \leftarrow S_t$ 
12:  else
13:     $I \leftarrow I + 1$ 
14:  if  $\text{Random}(0, 1) \leq \exp((f(S_t) - f(S_c))/T)$  (3.4)
15:     $S_c \leftarrow S_t$ 
16:  update the weight of the d and r according to their performance (3.3)
17:  if  $I \geq I_m$  then
18:     $S_c \leftarrow S_b$ 
19:    reset the scores of operators and  $I$ 
20:   $T \leftarrow T \times c$ 
21:  return  $S_b$ 

```

3. 实验评估

本文所有的算法均在配置 R7-4800 处理器, 32GB 运行内存, 64 位 Windows 11 操作系统的计算机上进行。运用的求解器为 Cplex12.9, ALNS 算法的编程在 Matlab 2021a 中完成。

3.1. 实验环境

由于卡车-无人机系统研究目前仍处于发展完善阶段, 各种不同假设下的配送系统并无完全适合本文进行实验, 故本文不同规模大小的不同算例主要采用依据实际生活区域进行随机生成方式产生, 且通过对于一些关键点的限制和随机排布来达到预期的算力效果, 如在生成前对顾客点类型按是否能被无人机服务进行划分, 从而近似替代农村真实环境中大山、河流等因素的替代, 以达到更真实环境的效果。

本文共生成 50 个 FVRPD 算例, 共 5 组顾客节点数, 为 $n = \{10, 25, 50, 75, 100\}$, 每组节点数各 10 个不同算例, 其中每组顾客点分布在 40×40 与 50×50 样例的各 5 个算例, 中小规模问题 $n = \{10, 25, 50\}$, 大规模问题 $n = \{75, 100\}$ 。对于所有的算例都有唯一卡车 K 与无人机 $D = 2$ 。所有的顾客点均可以被卡车服务, 但仅有 90% 的点可以被无人机服务。顾客点的需求 d_i 对于可以被无人机服务的顾客点在区间 $[0, 20]$ 上随机生成, 对于仅可被卡车服务的点则在区间 $[20, 100]$ 上随机生成[12]。假设卡车的容量 $Q^K = 3T$, 行驶的平均速度 $V^K = 35 \text{ km/h}$; 对于无人机, 最大载重 $Q^D = 30 \text{ kg}$, 平均速度 $V^D = 70 \text{ km/h}$ 。无人机最大续航时间 $e = 1 \text{ h}$ 。图 1 为已设置好参数的算例生成图。

3.2. 实验结果

本节将为围绕 50 个不同的算例展开实验计算结果的详述, 其中中小规模问题实验结果如下表 2 所示, 对于结果保留小数点后一位。表中分别展示了 ALNS 算法实验结果与运用求解器 Cplex 求解出的结果, 并综合对比了两种求解方法的解的质量与完成的时间。

表中第一步应用 Cplex 求解本文所提出的模型, 并将求解时间限制在 3600 s, 记录每个算例的下界 B_1 , 计算出每个算例的最优目标函数值 Z^C 及对应的运算时间 T^C ; 第二步使用本文提出的 ALNS 算法对小规模的每个算例进行计算, 为了更好地测试 ALNS 算法的稳定性, ALNS 算法均会对每个算例进行十次

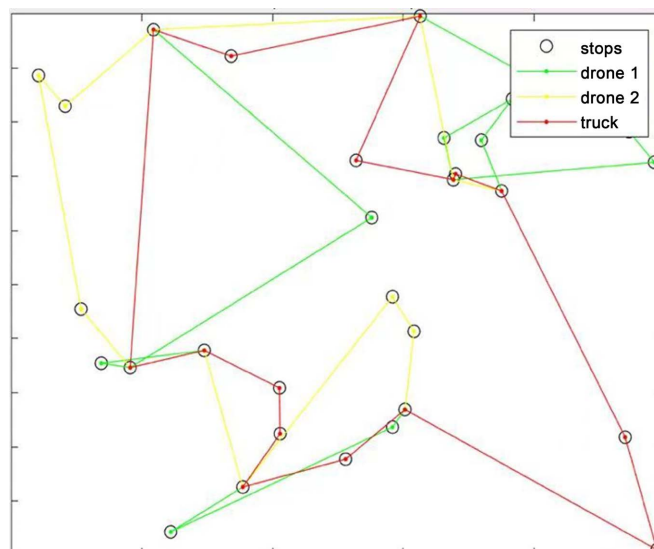


Figure 1. UAV truck system instance

图 1. 无人机 - 卡车系统算例示例

运算, 十次运算会得到一个平均最优目标函数值 $\bar{Z}^A$ 及一个平均运行时间 $\bar{T}^A$ 。此外表中还有三项衡量实验结果的指标 $\text{Gap}_C(\%)$ 、 $\text{Gap}_A(\%)$ 、 $\text{Imp}_A(\%)$ 。前两者描述了 Cplex 与 ALNS 算法距离下界的程度, 后者表示 ALNS 相较于 Cplex 求解的提升程度。实验计算结果如下表 1 实验结果汇总表。其中三项指标的表达式分别为:

$$\text{Gap}_C(\%) = \frac{Z^C - B_l}{Z^C} \times 100, \quad \text{Gap}_A(\%) = \frac{\bar{Z}^A - B_l}{Z^C} \times 100, \quad \text{Imp}_A(\%) = \frac{Z^C - \bar{Z}^A}{Z^C} \times 100。$$

详细计算结果如下表 2 实验结果汇总表。对于只有 10 个客户点的算例, Cplex 与本文 ALNS 算法均找到了 9/10 的最优解, 但 ALNS 算法的平均计算时间为 15.3 s, 比 Cplex 求解器计算平均花费时间 662.9 s 要小得多; 当客户点数提升至 25 与 50 后, 表中 $\text{Gap}_C(\%)$ 和 $\text{Gap}_A(\%)$ 已经不存在结果为 0 的情况, 说明此时两种方法均不能求解目标函数值至最优解, 但仍然可以发现此时除了 25 客户点下的第三组, 其他组的求解结果 ALNS 均优于 Cplex 求解器, 顾客点数为 25 下的 $\text{Imp}_A(\%)$ 的平均值达到了 10.5, 顾客点数升至 50 时 $\text{Imp}_A(\%)$ 则达到了 24.4。并且在客户点数量达到 50 之后, Cplex 求解器的求解时间均已达到最大时间 3600 s, 相比之下 ALNS 算法仅仅用到了 117.9 s 的平均时间。综上可以看出 ALNS 算法在中小型规模问题上较大程度上的提升了求解的质量与效率。

Table 2. Summary table of the experimental results

表 2. 中小规模实验结果汇总表

算例	B_l	Z^C	$\bar{Z}^A$	$\text{Gap}_C(\%)$	$\text{Gap}_A(\%)$	$\text{Imp}_A(\%)$	T^C	$\bar{T}^A$
10	1	52.1	52.1	0	0	0	59.5	14.4
	2	30.5	44.5	33.4	33.4	0	3600.0	15.3
	3	90.9	90.9	0	0	0	112.3	16.0
	4	75.7	75.7	0	0	0	1127.8	16.5
	5	88.2	88.2	0	0	0	409.1	13.2
	6	63.3	63.3	0	0	0	89.9	13.9

续表

	7	57.6	57.6	57.6	0	0	0	851.2	15.2
	8	55.8	55.8	55.8	0	0	0	160.9	13.9
	9	69.2	69.2	69.2	0	0	0	105.2	16.8
	10	77.4	77.4	77.4	0	0	0	113.2	15.5
均值		—	63.2	63.2	3.3	3.3	0	662.9	15.3
25	1	100.7	144.5	136.9	30.3	25.1	5.3	2450.2	38.6
	2	56.6	81.6	80.5	30.6	29.3	1.3	1852.3	50.2
	3	86.4	130.9	124.4	34.0	29.0	5.0	2863.5	40.8
	4	63.9	118.2	107.4	45.9	36.8	9.1	2536.4	45.3
	5	71.1	110.6	95.2	35.7	21.8	13.9	2753.2	55.2
	6	45.2	116.1	92.3	61.1	40.6	20.5	3205.2	41.3
	7	96.8	136.4	136.9	29.0	29.4	−0.4	1905.0	36.5
	8	88.5	147.6	140.3	40.0	35.1	4.9	2535.5	32.2
	9	35.2	104.2	83.7	66.2	36.9	19.7	2580.6	47.8
	10	106.2	182.3	153.9	41.7	26.2	15.6	1789.3	51.9
均值		—	123.5	113.5	41.5	31.0	10.5	2432.3	44.0
50	1	98.0	321.3	225.6	69.5	29.8	39.7	3600.0	113.9
	2	82.3	152.7	99.1	46.1	35.1	11.0	3600.0	115.8
	3	151.9	335.3	195.8	54.7	31.6	13.1	3600.0	119.4
	4	92.4	206.2	137.4	55.2	33.4	21.8	3600.0	113.4
	5	96.5	262.2	192.5	63.2	26.6	36.6	3600.0	120.1
	6	102.7	217.6	140.6	52.8	35.4	17.4	3600.0	124.2
	7	120.3	237.8	150.0	49.4	36.9	12.5	3600.0	115.2
	8	88.6	263.0	145.4	66.3	42.7	21.6	3600.0	123.6
	9	73.9	166.4	124.0	55.6	22.5	30.1	3600.0	110.6
	10	144.1	500.3	368.7	71.2	26.3	44.9	3600.0	122.6
均值			272.4	182.5	57.5	33.1	24.4	3600.0	120.2

对于较大规模问题 $n = \{75, 100\}$ 总共 20 数量的算例，ALNS 算法也同样采取实验十次求均值方式完成。详细计算结果如下表 3 大规模实验汇总表所示。

Table 3. Summary table of the experimental results of big-size
表 3. 大规模是实验结果汇总表

算例	n = 75				n = 100			
	Z^A	$\check{Z}^A$	Gap	$\check{T}^A$	Z^A	$\check{Z}^A$	Gap	$\check{T}^A$
1	178.6	182.3	2.0	532.2	230.3	236.7	2.7	1131.2
2	159.4	163.6	2.6	580.5	248.6	256.8	3.2	1032.7

续表

3	183.3	186.5	1.7	496.0	196.5	204.2	3.8	960.0
4	230.2	237.0	2.9	558.6	377.2	390.3	3.4	1010.5
5	198.4	203.4	2.5	592.6	297.4	308.2	3.5	1079.8
6	265.7	271.5	2.1	513.4	355.9	363.6	2.1	896.4
7	246.1	250.3	1.7	530.8	203.7	206.5	1.4	976.7
8	271.9	276.3	1.6	613.9	255.6	262.9	2.8	1089.2
9	228.3	231.4	1.3	546.7	278.1	288.7	3.7	1102.0
10	196.8	199.6	1.4	539.3	303.5	309.6	2.0	1040.1
均值	215.9	220.2	2.0	550.4	274.7	282.8	2.9	1031.9

由于问题规模的增大,此篇幅提到的模型要想获取更优解变得困难重重,甚至成为了 NP-hard 问题,且考虑到该问题类型并无完全对等的其他启发式算法进行比较,因此记录下述实验结果以及给出相对应指标来检验算法在较大规模问题中的发挥水平。首先对每组算例重复地运行十次,记录目标函数平均值 $\bar{Z}^A$ 、最优值 Z^A 以及平均运行时间 $\bar{t}^A$, 解的离散程度 $\text{Gap}(\%) = \frac{\bar{Z}^A - Z^A}{\bar{Z}^A} \times 100$ 。

伴随问题规模的逐渐增大,实验平均计算时间发生明显的改变,在顾客点为 75、100 时,ALNS 算法的平均时间 $\bar{t}^A$ 分别为 550.4、1031.9,但耗时仍处于可接受范围内,而 $\text{Gap}(\%)$ 的平均值分别为 2.0、2.9,其中 $\text{Gap}(\%)$ 的最小值为 1.3,最大值为 3.8,大规模问题通过 ALNS 算法所求得的目标值距离平均水平浮动较小,因此在求解大规模此问题时 ALNS 算法具备了稳定性和可行性。综上可以看出在大规模问题中,ALNS 算法同样能提供较高质量的解。

3.3. 系统优化分析

为验证本文无人机-卡车联合配送系统的高效性,引入三种不同配送方式并分别计算其最低成本:第一种方法单独卡车配送;第二种方法单独无人机配送,在所有客户都由从仓库派来的能提供无限电池容量的无人机条件下进行;第三为本文提出的卡车-无人机联合配送,无人机数量为 2。卡车行驶每千米的单位成本 $c^K = 1$,无人机行驶每千米的单位成本 $c^D = 0.2$ (图 2)。

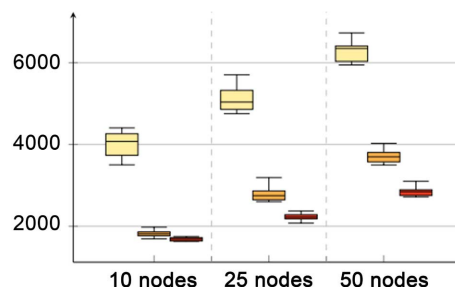


Figure 2. UAV truck system instance

图 2. 无人机-卡车系统算例示例

在本文实验环境的基础上分别求得三种方式在客户点数为 10、25、50 下求得的最低成本,其中黄色为方法三联合配送,橙色为方法二无人机配送,红色为方法一卡车配送。可以直观地看出卡车-无人机联合配送比其他两种方式节省了更多的成本。

4. 结论

本文提出了一种卡车 - 无人机联合配送系统并建立了对应模型, 这种模型中无人机可以携带多需求, 且能与卡车进行协同配送作业, 并相较于单独的卡车或无人机配送效率更高, 成本更低, 且摆脱了卡车无人机协同过程中等待时间的问题。将卡车作为无人机的发射载体, 使无人机在任意节点均能发射服务其他客户, 使得配送的整体过程更加灵活, 应对更复杂的环境, 卡车不方便到达的配送点可发射无人机进行配送, 同时卡车携带无人机又弥补了无人机续航不足的短板。

试验结果表明使用 ALNS 算法能够优化本文所提出的模型求解, 相较于 Cplex 求解器, 随着客户点及需求的增加放大, ALNS 算法也逐渐表现出更高效与高质量的求解, 当达到 50 客户点的中规模问题时, 可整体比 Cplex 求解质量提升 24.4%。并且在较大规模问题也能高效率高质量地提供更优目标值。

参考文献

- [1] Bektas, T. (2006) The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures. *Omega*, **34**, 209-219. <https://doi.org/10.1016/j.omega.2004.10.004>
- [2] Agatz, N., Bouman, P. and Schmidt, M. (2018) Optimization Approaches for the Traveling Salesman Problem with Drone. *Transportation Science*, **52**, 965-981. <https://doi.org/10.1287/trsc.2017.0791>
- [3] Cheng, C., Adulyasak, Y. and Rousseau, L. (2024) Robust Drone Delivery with Weather Information. *Manufacturing & Service Operations Management*, **26**, 1402-1421. <https://doi.org/10.1287/msom.2022.0339>
- [4] Murray, C.C. and Chu, A.G. (2015) The Flying Sidekick Traveling Salesman Problem: Optimization of Drone-Assisted Parcel Delivery. *Transportation Research Part C: Emerging Technologies*, **54**, 86-109. <https://doi.org/10.1016/j.trc.2015.03.005>
- [5] Schermer, D., Moeini, M. and Wendt, O. (2019) The Traveling Salesman Drone Station Location Problem. In: Le Thi, H., Le, H. and Pham Dinh, T., Eds., *Optimization of Complex Systems: Theory, Models, Algorithms and Applications*, Springer, 1129-1138. https://doi.org/10.1007/978-3-030-21803-4_111
- [6] Dayarian, I., Savelsbergh, M. and Clarke, J. (2020) Same-Day Delivery with Drone Resupply. *Transportation Science*, **54**, 229-249. <https://doi.org/10.1287/trsc.2019.0944>
- [7] 黄凯明, 卢才武, 连民杰. 三层级设施选址-路径规划问题建模及算法研究[J]. 系统工程理论与实践, 2018, 38(3): 743-754.
- [8] 李想, 李苏剑, 李宏. 两级选址-路径问题的大规模邻域搜索模拟退火算法[J]. 工程科学学报, 2017, 39(6): 953-961.
- [9] Boccia, M., Masone, A., Sforza, A. and Sterle, C. (2021) A Column-and-Row Generation Approach for the Flying Sidekick Travelling Salesman Problem. *Transportation Research Part C: Emerging Technologies*, **124**, Article ID: 102913. <https://doi.org/10.1016/j.trc.2020.102913>
- [10] Poikonen, S. and Golden, B. (2020) The Mothership and Drone Routing Problem. *INFORMS Journal on Computing*, **32**, 249-262. <https://doi.org/10.1287/ijoc.2018.0879>
- [11] Poikonen, S. and Golden, B. (2020) Multi-Visit Drone Routing Problem. *Computers & Operations Research*, **113**, Article ID: 104802. <https://doi.org/10.1016/j.cor.2019.104802>
- [12] Wang, Z. and Sheu, J. (2019) Vehicle Routing Problem with Drones. *Transportation Research Part B: Methodological*, **122**, 350-364. <https://doi.org/10.1016/j.trb.2019.03.005>