

基于自适应大规模邻域搜索算法的多车辆与 多无人机协同配送方法

伍国华¹, 毛妮¹, 徐彬杰¹, 马中强¹, 赵苛欣¹, 李志猛^{2†}

(1. 中南大学 交通运输工程学院, 长沙 410075; 2. 国防科技大学 系统工程学院, 长沙 410073)

摘要: 针对物流配送需求大、“最后一公里”交付困难等问题,提出带有动态能耗约束的多车辆与多无人机协同配送问题,并以最小化配送时间为目标建立混合整数规划模型(MIP).为解决该问题,设计 K -means 聚类 and 最近邻协同的初始解生成算法,并提出基于问题领域知识的自适应大规模邻域搜索算法(adaptive large neighborhood search, ALNS).在不同规模算例上的实验结果表明,所提出的算法相比于模拟退火算法、变邻域搜索算法和遗传算法在求解质量和求解效率方面都具有一定的优势,求解质量分别平均提升 23.8%、23.3% 和 5.7%,表明 ALNS 较对比算法能够更好地平衡全局搜索和局部搜索.此外,灵敏度分析实验表明,无人机载重能力和无人机续航能力是影响包裹配送时间的两个关键因素.

关键词: 车辆与无人机; 协同配送; 整数规划; 动态能耗; 路径规划; 自适应大规模邻域搜索

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.2268

引用格式: 伍国华,毛妮,徐彬杰,等.基于自适应大规模邻域搜索算法的多车辆与多无人机协同配送方法[J].控制与决策,2023,38(1): 201-210.

The cooperative delivery of multiple vehicles and multiple drones based on adaptive large neighborhood search

WU Guo-hua¹, MAO Ni¹, XU Bin-jie¹, MA Zhong-qiang¹, ZHAO Ke-xin¹, LI Zhi-meng^{2†}

(1. School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China; 2. School of Systems Engineering, National University of Technology, Changsha 410073, China)

Abstract: To solve the problem of huge distribution demand and “last mile” distribution, this paper first proposes the cooperative delivery of multiple vehicles and multiple drones with dynamic energy consumption (CDMVMD-DEC), and provides a mixed integer programming model (MIP) aimed at minimizing the delivery time. To solve the problem efficiently, the adaptive large neighborhood search (ALNS) based on problem domain knowledge is proposed, along with the combination of the K -means clustering and the nearest neighbor for constructing the initial solution. Experiments on different-scale instances demonstrate that the ALNS outperforms the simulated annealing, variable neighborhood search and genetic algorithm in solution quality and computational time. In terms of solution quality, the performance of the ALNS is improved by 23.8%, 23.3% and 5.7% respectively. The results of experiments show that ALNS provides a better balance between global search and local search. Moreover, the results of the sensitivity test show that the load capacity and endurance of drone are the important factors affecting the delivery time.

Keywords: multiple vehicles and multiple drones; cooperative delivery; integer programming; dynamic energy consumption; route planning; adaptive large neighborhood search

0 引言

随着科技不断发展,无人机应用越来越广泛,尤其在物流领域更是备受关注.亚马逊公司^[1]、DHL 公司^[2]、Google 公司^[3]等都陆续开展了无人机物流配送

的测试与应用.2020年爆发的全球性公共卫生事件将“无接触物流”配送推向前台^[4],无人机作为“无接触物流”配送的重要工具受到了更为广泛的关注.无人机配送模式高效、环保,具有巨大的发展前

收稿日期: 2021-12-29; 录用日期: 2022-04-15.

基金项目: 国家自然科学基金面上项目(62073341).

责任编辑: 樊治平.

[†]通讯作者. E-mail: zmli@nudt.edu.cn.

重力加速度, ρ 为空气密度, ζ 为旋转叶片圆盘面积, n 为无人机转子数量. 因此, 无人机的能耗与飞行时间和装载包裹重量相关, 其公式为

$$E = p(m) \cdot t. \quad (2)$$

其中: E 为无人机电池的总能量, $p(m)$ 为电池输出功率, t 为飞行时间. 根据问题描述设定符号和变量如表1所示.

表1 符号及其说明

符号	说明	符号	说明
V	服务节点, $V = \{0, 1, \dots, n, n+1\}$, 其中0和 $n+1$ 为仓库节点	v_t	货车的行驶速度
C	客户点, $C = \{1, 2, \dots, n\}$	v_d	无人机的飞行速度
m_i	客户点 i 的包裹重量	t_j^k	货车 $k \in \Omega$ 到达服务节点 j 的时间
w_d	无人机自重	q_i^k	货车 $k \in \Omega$ 从服务节点 i 出发的时间
Q	无人机的最大载荷重量	$f_j^{u,k}$	从货车 $k \in \Omega$ 起飞的无人机 $u \in \Upsilon$ 到达服务节点 j 的时间
E	无人机电池最大能量	$a_i^{u,k}$	决策变量, 为0表示货车 $k \in \Omega$ 上的无人机 $u \in \Upsilon$ 在服务节点 i 处被发射, 否则为1
$d_{i,j}$	i 点到 j 点的距离	$b_i^{u,k}$	决策变量, 为0表示货车 $k \in \Omega$ 上的无人机 $u \in \Upsilon$ 在服务节点 i 处被回收, 否则为1
$x_{i,j}^k$	决策变量, 为1表示货车 $k \in \Omega$ 从 i 点行驶至 j 点, 否则为0	$w_j^{u,k}$	从货车 $k \in \Omega$ 上起飞的无人机 $u \in \Upsilon$ 到达服务节点 j 时的剩余载重量
$y_{i,j}^{u,k}$	决策变量, 为1表示从货车 $k \in \Omega$ 起飞的无人机 $u \in \Upsilon$ 从 i 点飞行至 j 点, 否则为0	$e_j^{u,k}$	从货车 $k \in \Omega$ 上起飞的无人机 $u \in \Upsilon$ 到达服务节点 j 时剩余能量
$s_i^{u,k}$	决策变量, 为1表示货车 $k \in \Omega$ 上无人机 $u \in \Upsilon$ 能够在服务节点 i 处放飞无人机, 否则为0	M	足够大的正数

1.2 数学模型

1) 目标函数.

车辆与无人机协同配送问题的目标是使总配送时间最小化, 构建目标函数为

$$\min\{t | t \geq t_{n+1}^k, k \in \Omega\}. \quad (3)$$

2) 配送约束.

$$\sum_{k \in \Omega} \sum_{j \in C_+} x_{i,j}^k + \sum_{k \in \Omega} \sum_{u \in \Upsilon} \left(a_i^{u,k} \sum_{j \in C_+} y_{i,j}^{u,k} \right) = 1, \quad \forall i \in C; \quad (4)$$

$$\sum_{k \in \Omega} \sum_{i \in C_0} x_{i,j}^k + \sum_{k \in \Omega} \sum_{u \in \Upsilon} \left(b_j^{u,k} \sum_{i \in C_0} y_{i,j}^{u,k} \right) = 1, \quad \forall j \in C; \quad (5)$$

$$\sum_{i \in C} x_{0,i}^k = 1, \quad \forall k \in \Omega; \quad (6)$$

$$\sum_{i \in C} x_{i,n+1}^k = 1, \quad \forall k \in \Omega. \quad (7)$$

式(4)和(5)表示每个客户点只能被服务一次, 无人机的发射点和回收点由车辆进行配送服务. 式(6)和(7)保证所有车辆只从仓库出发一次, 最终返回一次.

3) 协同约束.

$$\sum_{j \in C_+} y_{i,j}^{u,k} \leq \sum_{v \in C_+} x_{i,v}^k, \quad \forall k \in \Omega, u \in \Upsilon,$$

$$i \in \{C_0, i \neq j, i \neq v\}; \quad (8)$$

$$\sum_{i \in C_0} y_{i,j}^{u,k} \leq \sum_{v \in C_0} x_{v,j}^k, \quad \forall k \in \Omega, u \in \Upsilon,$$

$$j \in \{C_+, j \neq i, j \neq v\}; \quad (9)$$

$$(1 - a_i^{u,k}) \left(1 - \sum_{j \in C_+} y_{i,j}^{u,k} \right) = 0,$$

$$\forall i \in C_0, k \in \Omega, u \in \Upsilon; \quad (10)$$

$$(1 - a_i^{u,k}) \left(1 - \sum_{j \in C_+} x_{i,j}^k \right) = 0,$$

$$\forall i \in C_0, k \in \Omega; \quad (11)$$

$$(1 - b_j^{u,k}) \left(1 - \sum_{i \in C_0} y_{i,j}^{u,k} \right) = 0,$$

$$\forall j \in C_+, k \in \Omega, u \in \Upsilon; \quad (12)$$

$$(1 - b_j^{u,k}) \left(1 - \sum_{i \in C_0} x_{i,j}^k \right) = 0,$$

$$\forall j \in C_+, k \in \Omega. \quad (13)$$

式(8)、(10)和(11)保证了若无人机从 i 点起飞, 则该点必然是车辆与无人机访问过的路径点; 式(9)、(12)和(13)保证了若无人机在 j 点降落, 则该点必然是车辆与无人机访问过的路径点.

4) 多架无人机调度约束.

$$\sum_{u \in \Upsilon} \left[(1 - a_i^{u,k}) \sum_{j \in C_+} y_{i,j}^{u,k} \right] \leq U,$$

$$\forall i \in C_0, i \neq j, k \in \Omega; \quad (14)$$

$$\sum_{u \in \mathcal{U}} \left[(1 - b_i^{u,k}) \sum_{i \in C_0} y_{i,j}^{u,k} \right] \leq U,$$

$$\forall j \in C_+, i \neq j, k \in \Omega; \quad (15)$$

$$(1 - a_i^{u,k})(1 - s_i^{u,k}) = 0, \forall i \in C_0, k \in \Omega, u \in \mathcal{U}; \quad (16)$$

$$(1 - b_i^{u,k})(1 - s_i^{u,k}) = 0, \forall i \in C_+, k \in \Omega, u \in \mathcal{U}; \quad (17)$$

$$x_{i,j}^k + (1 - a_i^{u,k}) + b_j^{u,k} + s_j^{u,k} \leq 3,$$

$$\forall i \in C_0, j \in C_+, i \neq j, k \in \Omega, u \in \mathcal{U}; \quad (18)$$

$$x_{i,j}^k + (1 - s_i^{u,k}) + b_j^{u,k} + s_j^{u,k} \leq 3,$$

$$\forall i \in C_0, j \in C_+, i \neq j, k \in \Omega, u \in \mathcal{U}. \quad (19)$$

式(14)确保在车辆服务点处发射的无人机数量不得超过无人机总数;同样地,式(15)确保回收点处的无人机数量也不得超过无人机总数;式(16)表示若*i*点为无人机*u*发射点,则该点可以发射无人机*u*;式(17)表示若*i*点为无人机*u*的回收点,则该点可以继续发射无人机*u*;式(18)表示若车辆从*i*点行驶至*j*点,且*i*点为无人机*u*的起飞点,在*j*点处无人机*u*仍未被回收,则*j*点不能继续发射无人机*u*;式(19)表示若车辆从*i*点行驶至*j*点,且无人机*u*还未回到*i*、*j*点,则*j*点不可继续发射无人机*u*.

5) 时间约束.

$$t_0^k, q_0^k = 0, \forall k \in \Omega; \quad (20)$$

$$q_i^k \geq t_i^k, \forall i \in C_0, k \in \Omega; \quad (21)$$

$$q_i^k \geq f_i^{u,k} - M \cdot b_i^{u,k}, \forall i \in C_+, k \in \Omega, u \in \mathcal{U}; \quad (22)$$

$$t_j^k \geq q_i^k + \frac{d_{i,j}}{v_t} - M(1 - x_{i,j}^k), \forall i \in C_0,$$

$$j \in \{C_+ : i \neq j\}, k \in \Omega; \quad (23)$$

$$(1 - a_i^{u,k})(t_i^k - f_i^{u,k}) = 0, \forall i \in C_0, k \in \Omega, u \in \mathcal{U}; \quad (24)$$

$$f_j^{u,k} \geq f_i^{u,k} + \frac{d_{i,j}}{v_d} - M(1 - y_{i,j}^{u,k}), \forall i \in C_0,$$

$$j \in \{C_+ : i \neq j\}, k \in \Omega, u \in \mathcal{U}. \quad (25)$$

式(20)设置车辆在初始仓库0的出发和到达时间为0;式(21)和(22)表示在无人机回收点*i*处,车辆需等待车辆与无人机会合后再出发,因此车辆在回收点*i*的出发时间大于等于车辆到达时间和无人机到达时间;式(23)保证了车辆到达*j*点的时间大于等于从上一个点*i*出发的时间与两点间的行驶时间之和;式(24)表示无人机在起飞点处的出发时间等于车辆到达该点的时间;式(25)约束无人机到达*j*点的时

间要大于或等于从上一个点*i*出发的时间与两点间的行驶时间之和.

6) 无人机载重约束.

$$w_i^{u,k} \geq 0, \forall i \in V, k \in \Omega, u \in \mathcal{U}; \quad (26)$$

$$(1 - b_i^{u,k})(w_i^{u,k} - Q) = 0, \forall i \in C_+, k \in \Omega, u \in \mathcal{U}; \quad (27)$$

$$w_j^{u,k} \geq w_i^{u,k} + m_i - M(1 - y_{i,j}^{u,k}), \forall i \in C_0,$$

$$j \in \{C_+ : i \neq j\}, k \in \Omega, u \in \mathcal{U}. \quad (28)$$

式(26)保证无人机在任意一点的剩余载重量均大于或等于0;式(27)表示在无人机回收点处,无人机的剩余载重量为其最大载重能力;式(28)表示无人机在*j*的剩余载重量大于等于上一客户点*i*的剩余载重量加上该点的包裹重量.

7) 无人机能量约束.

$$e_i^{u,k} \geq 0, \forall i \in V, k \in \Omega, u \in \mathcal{U}; \quad (29)$$

$$(1 - a_i^{u,k})(e_i^{u,k} - E) = 0, \forall i \in C_0, k \in \Omega, u \in \mathcal{U}; \quad (30)$$

$$e_j^{u,k} \leq e_i^{u,k} - p(w_i^{u,k}) \left(\frac{d_{i,j}}{v_d} \right) + M(1 - y_{i,j}^{u,k}),$$

$$\forall i \in C_0, j \in \{C_+ : i \neq j\}, k \in \Omega, u \in \mathcal{U}. \quad (31)$$

式(29)保证无人机在任意一点的剩余能量均大于或等于0;式(30)确保在无人机起飞处,其剩余能量为电池的最大能量;式(31)保证无人机到达*j*点的能量要小于到达上一客户点*i*的能量.

2 求解算法

车辆与无人机协同配送问题属于NP-hard^[13],采用精确算法求解大规模实例会出现运行时间过长的的问题,因此,本文提出自适应大规模邻域搜索算法求解该问题. ALNS是一种元启发式算法,它在大邻域搜索的基础上为破坏算子与修复算子赋予权重,使得算法能够自动选择表现较好的算子对解进行改进,从而在合理的运行时间内求得更好的解,算法还利用Metropolis接受准则避免陷入局部最优^[14].

本文提出的ALNS以破坏算子与修复算子成对组合形式对解进行重构,每对算子对的权重记为 ρ_j . 算法根据各算子对的权重,利用轮盘赌原理选择出算子对. 算子对*j*被选择的概率^[14]为

$$\phi_j = \frac{\rho_j}{|\Omega|}, \quad j = 1, 2, \dots, |\Omega|. \quad (32)$$

$$\sum_{k=1} \rho_k$$

算子对的权重根据迭代过程中的表现评分变化实现动态调整. 当算子对搜索后产生的新解*s**是新的全局最优解时,赋予 w_1 分;当新解*s**比当前解*s*好

时,赋予 w_2 分;当新解 s^* 比当前解 s 差但被接受准则接受时,赋予 w_3 分,否则,赋予 w_4 ,其中 $w_1 \geq w_2 \geq w_3 \geq w_4$. 权重 ρ_j 的更新公式为

$$\rho_j = \lambda \rho_j^0 + (1 - \lambda) \psi_j, j = 1, 2, \dots, |\Omega|. \quad (33)$$

其中: λ 为反应参数, $\lambda \in [0, 1]$; ψ_j 为算子对 j 的分数; ρ_j^0 为历史权重.

ALNS 的具体实现过程如下.

step 1: 初始化最优解 s_{best} 和当前解 s 为初始解 s_0 , 设置 Metropolis 接受准则中的初始温度 T_0 、最终温度 T_{min} 、温度冷却率 γ 以及最大迭代次数 Max_{it} , 并将破坏算子与修复算子的权重 ρ 初始化为相同的权重, 迭代次数 it 初始化为 1, 当前温度 T 初始化为初始温度 T_0 .

step 2: 依据算子对权重 ρ 从邻域集合 Ω 中选择一对破坏与修复算子 $d\&r$, 生成新解 s^* .

step 3: 比较新解 s^* 与当前解 s , 若新解 s^* 比当前解 s 好, 则将当前解 s 更新为新解 s^* , 否则将新解 s^* 交给 Metropolis 接受准则进行判断. 将更新后的当前解 s 与最优解 s_{best} 进行比较, 如果当前解 s 优于最优解 s_{best} , 则最优解 s_{best} 会被更新, 否则不变.

step 4: 每 x 代作为一个记分区间, 记录区间内各算子对的得分情况. 当一个记分区间结束时, 根据式 (33) 调整各算子对的权重.

step 5: 重复执行 step 2~step 4, 直到迭代次数 it 达到最大迭代次数 Max_{it} 或温度 T 达到最低温度 T_{min} 时终止, 即得到全局最优解 s_{best} .

2.1 初始解

初始解的生成遵循了先构造车辆路由, 再以配送时间最短为目标插入无人机路由的思想^[1], 本文据此设计了 K -means 聚类 and 最近邻结合的初始解算法.

在车辆路径的生成中, 首先将簇团数 k_1 初始化为车辆数 k , 使用 K -means 聚类算法^[15] 对所有客户点进行聚类, 每一个簇团内的客户点即为每辆车的服务范围. 因为 K -means 聚类算法以样本点之间的距离大小作为划分簇团的依据, 所以划分在一个簇团内的点距离较近, 便于车辆配送包裹至客户点. 在每一个簇团内, 通过最近邻算法^[16] 生成车辆路径.

在车辆路由生成的基础上, 对每一条车辆路由上的客户点再进行一次 K -means 聚类以确定无人机的服务范围, 每一个簇团内的客户点数应为多架无人机配送点的数量. 对于第 2 次 K -means 聚类中簇团数 k_2 的取值, 考虑到配送无人机平均最大载重量为 3 kg, 每个包裹的重量约为 1 kg^[17], 因此每架无人机平均可以携带 3 个包裹, 每段无人机路由平均包含 5 个

点: 3 个配送点和 2 个起降点. 因此, 簇团数 k_2 由 $k_2 = N_k / (u \times 5)$ 确定, 其中 N_k 为第 k 条车辆路由中客户点总数, u 为第 k 辆车上携带的无人机数量.

在每一个簇团内, 选定车辆路由中的第 1 个点作为无人机发射点, 再次通过最近邻算法确定每架无人机的路由, 直至无可配送的无人机或簇团内的点均不满足无人机配送的载重或者续航能力约束为止.

2.2 邻域结构及解的调整

ALNS 的邻域结构由破坏算子和修复算子构成, 因此本文设计 4 种破坏算子和 4 种修复算子. 破坏算子与修复算子将以成对的形式出现, 一对即为一种邻域结构.

2.2.1 邻域结构

破坏算子 d_1 : 随机移除 1 个服务点;

破坏算子 d_2 : 随机移除 2 个服务点;

破坏算子 d_3 : 随机移除 1 个无人机路径中的起飞点(或降落点);

破坏算子 d_4 : 随机移除 2 个车辆配送点间的路段;

修复算子 r_1 : 将随机移除的 1 个服务点插入到 1 个随机路段内;

修复算子 r_2 : 将移除的两点交换位置, 再重新插入;

修复算子 r_3 : 将移除的无人机起飞点(或降落点)插入到 1 个车辆配送点的位置上;

修复算子 r_4 : 将随机移除的两个点之间的路段翻转, 重新插入到路径中.

对以上破坏和修复算子组合, 形成以下邻域:

1) 破坏算子 d_1 & 修复算子 r_1 .

该算子从当前路径中随机移除一点, 再将其重新插入至一个随机路段内. 使用该算子对后发生无人机的路径变化, 需检验涉及该操作的无人机路径是否满足载重和续航能力约束, 在不满足约束时进行解的调整. 如图 2 所示, 随机选定的车辆配送点 4 被重新插入到无人机路段 3-2 时, 此时无人机路径的降落点

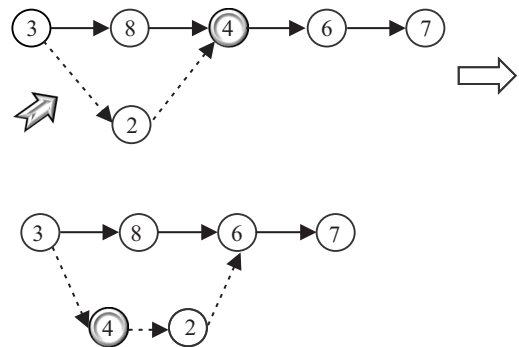


图 2 d_1 & r_1 算子对示意图

会改变,则选取下一个车辆配送点6作为无人机的降落点,无人机路段3-2中插入车辆配送点4.

2) 破坏算子 d_2 &修复算子 r_2 .

该算子对随机移除两个服务点交换位置后再插入,当两点中任意一点涉及无人机路径时,同样需进行无人机载重量和续航能力约束的检验.如图3所示,即为 d_2 & r_2 算子对示意图.

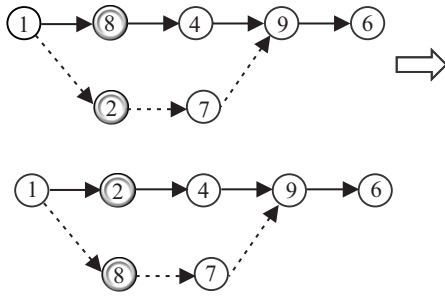


图3 d_2 & r_2 算子对示意图

3) 破坏算子 d_3 &修复算子 r_3 .

该算子对随机移除一个无人机路径中的起飞点(或降落点),再设置到新的车辆配送点上,此操作涉及无人机路径的变化,需进行无人机路径的载重量和续航能力约束检验.如图4所示,随机移除无人机路径8-1-6中的起飞点8,再插入在车辆配送点3的位置处.

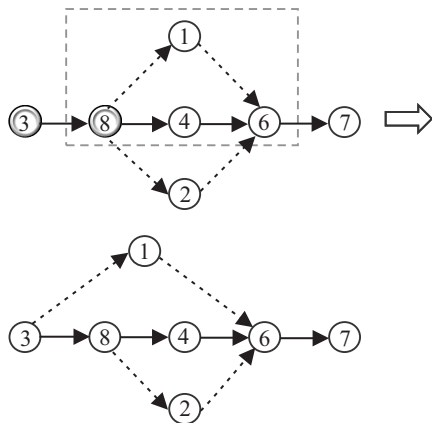


图4 d_3 & r_3 算子对示意图

4) 破坏算子 d_4 &修复算子 r_4 .

该算子对指随机移除两车辆配送点之间的路径,将两点之间的路径翻转其编号后插入到路径中,此操作可减少车辆路径交叉.如图5所示,随机移除车辆路径6-4-2-8,将点6与8之间的路径编号翻转,将新的路径6-2-4-8重新插入到路径中.

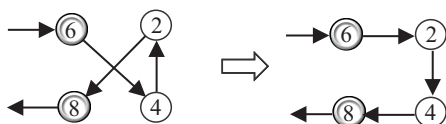


图5 d_4 & r_4 算子对示意图

2.2.2 解的调整

因无人机载重和续航能力有限,邻域搜索可能会产生不可行解,故本文设计了解的调整策略,即减少无人机在一段航程中配送任务量.具体操作是将无人机航程中的最后一个配送点更改为降落点插入车辆路径中,若仍不可行则继续使用操作直至满足约束为止.如图6所示,邻域搜索后生成的无人机路径1-8-7-9不可行,则将最后一个配送点7更改为降落点,产生新的无人机路径(如图6中解的调整1所示),再进行解的可行性判断.若仍不可行则继续将无人机路径的最后一个配送点8插入到车辆路径中,此时无人机路径将被取消(如图6中解的调整2所示).

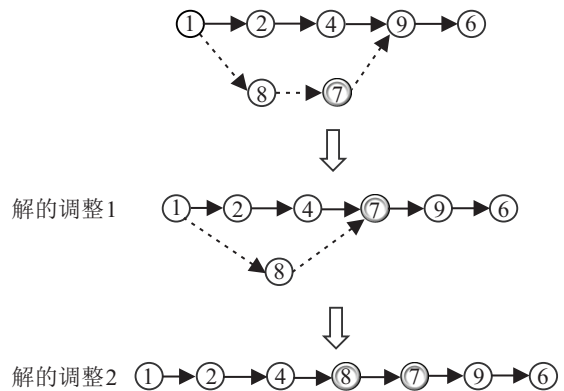


图6 无人机路径调整示例

3 实验分析

本文对ALNS在不同规模的算例上进行了性能评估,与Gurobi求解器得到的最优解进行比较,并与传统的模拟退火算法(SA)^[18]、传统的变邻域搜索算法(VNS)^[19]和遗传算法(GA)^[20]进行对比研究.所有实验在配置为Intel(R) Core(TM) i7-10700F CPU@2.90 GHz,内存为32 G的计算机上进行,所有算法和测试程序均由Matlab R2018a编程实现,为避免随机因素在每个实例上独立运行20次,配送时间即为20次运行结果的平均值(单位s).

3.1 算例及参数设置

为了验证算法在不同规模算例中的性能,采用文献[21-22]中车辆路径问题的标准算例,并对算例中的客户需求数据即包裹重量进行一定的调整,以满足80%的包裹重量小于2.3 kg的标准^[1].此外,根据车辆与无人机配送的发展现状及相关研究^[1,12],规定了本实验中采用的车辆和无人机的相关参数(见表2),实验采用六旋翼小型无人机进行配送服务,飞行中的空气流体密度为1.2 kg/m³^[12].

对比算法的初始解和邻域结构采用所提出的初始解算法和邻域结构,其他算法参数如表3所示.

表2 车辆和无人机的相关参数

车辆	速度 最大载荷重量	50 km/h ∞
无人机	速度 最大载荷重量 自重 电池能量 转子盘面积	75 km/h 3 kg 1.5 kg 500 J 0.2 m ²

表3 算法相关参数

	SA	VNS	GA	ALNS
最大迭代次数	—	1 000	1 000	1 000
初始温度 T_0	0.01	—	—	0.01
降温系数 γ	0.98	—	—	0.98
最低温度 T_{min}	0.001	—	—	0.001
变异概率 p_m	—	—	0.2	—
种群规模 M	—	—	10	—
反应参数 λ	—	—	—	0.9
权重分数 ω	—	—	—	$\omega_1 = 55, \omega_2 = 30,$ $\omega_3 = 15, \omega_4 = 10$

3.2 算法性能验证

本实验在小规模和大规模客户点算例中对ALNS求得的解与Gurobi求解器得到的解进行比较. 对于小规模算例, 实验修改了标准算例, 将客户节点减少到8~12, 安排2辆车、每辆车携带2架无人

机进行实验比较; 对于大规模算例, 安排2辆车、每辆车携带3架无人机, 设定Gurobi可接受运行时间为1 800 s, 对结果进行比较, 实验结果记录于表4.

表4 ALNS算法与Gurobi求解的对比结果

算例	客户数	Gurobi		ALNS		GAP	
		配送时间/s	运行时间/s	配送时间/s	运行时间/s		
小规模	P1-n9-k2	8	4 725.3	1 531.5	4 805.4	27.4	-1.7%
	P2-n9-k2	8	4 678.1	2 632.4	4 946.6	29.0	-5.7%
	P1-n11-k2	10	5 014.8	1 633.7	5 098.6	29.5	-1.7%
	P2-n11-k2	10	5 216.7	3 982.1	5 216.7	29.6	0.0%
	P1-n13-k2	12	4 887.2	2 428.5	4 956.6	30.2	-1.4%
大规模	P-n21-k2	20	9 543.2	1 800	7 640.2	32.3	19.9%
	A-n32-k5	31	28 246.8	1 800	19 999.6	39.7	29.2%
	A-n36-k5	35	38 592.5	1 800	20 444.9	42.4	47.0%
	P-n40-k5	39	24 957	1 800	12 451.4	41.7	50.1%
	P-n51-k10	50	31 520.2	1 800	14 382.2	49.5	54.4%
	P-n60-k10	59	43 851.5	1 800	17 289.5	46.8	60.6%

此外, 实验还设置了SA、VNS和GA作为对比算法, 选取20组不同规模的算例, 在2辆货车, 每辆货车上携带3架无人机的情况下进行实验. 为了清晰地呈现实验结果, 引入GAP值^[23]表示对比算法与ALNS的差距, GAP的计算方式如下所示:

$$GAP = \frac{T_X - T_{ALNS}}{\max\{T_X, T_{ALNS}\}} \quad (34)$$

其中: T_X 为Gurobi、SA、VNS或GA求得的解, T_{ALNS} 为ALNS求得的配送时间. 实验还进行了Wilcoxon有符号秩检验^[24], 其中 R^+ 表示在给定问题上当前算法优于第2种算法的秩之和, R^- 表示第2种算法优于当前算法的秩之和. 将ALNS与Gurobi求解的实验结果记录于表4, ALNS与4种对比算法结果记录于表5.

由表4可知, ALNS在所有小规模算例上表现较好, 与最优解相差0~5.7%, 且ALNS的运行时间仅需30 s, 比Gurobi消耗的计算时间少数倍. 这主要是因为模型复杂度较高, Gurobi的求解时间会随算例规模的增大而增加. 在大规模算例上, ALNS在运行时间和求解质量上都具有一定优势, 在设定Gurobi求

解1 800 s的前提下, ALNS可以在50 s以内求得满意解, 超出Gurobi求解结果的19.9%~60.6%.

在与其他对比算法的比较中, 根据表5可得, 在解的质量方面: ALNS算法在各种规模的算例中运行得到的解均优于3种对比算法. 在具有20个客户点的小规模算例中, 所有算法的解的质量相近, 这是因为客户规模较小, 算法求得的满意解都接近于最优解. 在具有40个客户点以上的大、中规模算例中, ALNS算法的优势较显著. 与SA相比, ALNS的求解质量提升2.3%~40.1%, 尤其在大、中规模算例中可提升27.0%以上; 与VNS相比, ALNS的求解质量平均提升23.1%; 与GA相比, ALNS的求解质量提升0.4%~15.1%. 因此, ALNS相比对比算法较好地平衡了局部搜索能力和全局搜索能力. 在Wilcoxon有符号秩检验中, 3种对比算法的 R^- 值为210, R^+ 值为0, 表明在所有算例中, 相比3种对比算法, ALNS能够取得更优的解, p 值显示在95%的置信度水平下, ALNS算法的优化性能显著优于对比算法.

表5 4种算法对比结果

算例	客户数	SA			VNS			GA			ALNS	
		配送时间/s	运行时间/s	GAP	配送时间/s	运行时间/s	GAP	配送时间/s	运行时间/s	GAP	配送时间/s	运行时间/s
P-n21-k2	20	7 820.1	30.1	2.3 %	7 600.4	61.6	-0.5 %	7 711.3	64.7	0.9 %	7 640.2	32.3
P-n23-k8	22	7 674.7	30.4	10.9 %	7 895.5	61.3	13.4 %	8 054.8	59.9	15.1 %	6 836.8	33.7
A-n32-k5	31	24 613.0	36.2	18.7 %	27 858.2	68.2	28.2 %	20 481.9	73.6	2.4 %	19 999.6	39.7
A-n36-k5	35	26 927.2	36.4	24.1 %	26 803.7	63.9	23.7 %	20 630.7	76.9	0.9 %	20 444.9	42.4
A-n38-k5	37	27 932.5	35.9	34.9 %	24 193.5	71.0	24.8 %	18 379.4	78.9	1.0 %	18 192.6	41.3
P-n40-k5	40	16 547.1	35.4	24.8 %	15 423.7	60.2	19.3 %	12 857.7	73.5	3.2 %	12 451.4	41.7
A-n44-k6	43	27 668.9	41.0	8.6 %	28 816.9	65.3	12.3 %	25 394.9	80.8	0.4 %	25 283.3	42.5
A-n46-k7	45	31 424.8	36.5	33.8 %	27 244.0	76.4	23.6 %	22 092.3	79.2	5.8 %	20 812.2	41.9
A-n48-k7	47	31 560.5	38.0	27.2 %	29 955.0	68.3	23.3 %	24 580.3	80.9	6.6 %	22 967.7	42.1
P-n51-k10	50	20 670.0	42.8	30.4 %	19 835.8	83.3	27.5 %	15 304.9	92.3	6.0 %	14 382.2	49.5
P-n55-k7	54	21 599.7	42.6	26.3 %	23 049.6	74.6	30.9 %	16 359.3	84.1	2.7 %	15 921.1	46.9
P-n60-k10	59	21 099.8	42.2	18.1 %	2 3084.5	77.4	25.1 %	18 186.9	81.3	4.9 %	17 289.5	46.8
A-n61-k9	60	32 871.1	43.8	40.1 %	31 670.4	78.6	37.8 %	20 250.0	94.8	2.7 %	19 705.2	51.1
A-n63-k9	62	39 336.7	44.5	27.9 %	35 178.8	83.0	19.4 %	29 416.5	97.1	3.6 %	28 366.5	50.4
P-n65-k10	64	23 094.2	43.4	17.6 %	25 868.1	88.0	26.4 %	20 787.8	80.2	8.4 %	19 037.3	46.4
A-n69-k9	68	41 782.5	44.4	38.1 %	42 474.4	87.6	39.1 %	26 355.1	100.0	1.9 %	25 859.6	50.7
P-n70-k10	69	27 499.5	43.2	23.7 %	24 036.9	91.3	12.7 %	21 256.9	82.7	1.3 %	20 972.2	45.5
P-n76-k4	75	28 110.3	47.2	23.0 %	29 362.3	96.3	26.3 %	22 266.7	93.0	2.8 %	21 634.4	51.9
A-n80-k10	80	47 601.2	50.9	31.1 %	45 260.8	96.9	27.6 %	33 597.8	110.1	2.4 %	32 780.9	58.2
P-n101-k4	100	28 066.6	57.2	13.5 %	33 098.4	110.5	26.6 %	24 435.3	119.7	0.6 %	24 290.4	62.9
Wilcoxon 符号秩检验	R^+	0			0			0			210	
	R^-	210			210			210			0	
	p 值	0.000082			0.000082			0.000082			0.000082	

随着算例规模的增加,在求解时间方面,各算法求解时间普遍增加,但ALNS几乎都能在较短的时间内求得更好的解. ALNS比SA运行时间稍长一些,平均差值为4.8 s,这是因为ALNS拥有多样化的领域结构和搜索空间,但其仍能够在30~60 s内获得比SA质量更优的解;ALNS比VNS运行时间平均短32.3 s,尤其在80个客户及以上的大规模算例中,ALNS可在60 s左右求得满意解,而VNS需要100 s以上;ALNS比GA运行时间平均短39.3 s,尤其在80个客户及以上的大规模算例中,GA需要运行110 s及以上.

3.3 收敛性分析

为了验证算法的收敛性,4种算法在不同规模的算例上的结果如图7所示. 由图7可见,ALNS能以较快的速度收敛到更优的解,这是因为ALNS较好地兼顾了搜索过程历史信息(自适应机制)和解的多样性(Metropolis接受准则),因此ALNS在不同规模的实例上能以相对较少的时间花费寻找到更优的解,在对比算法中具有较强的竞争优势. 图7(a)中,与3种对比算法相比,ALNS能够收敛至更好的解,VNS虽然收敛速度较快,但陷入了局部最优;图7(b)中,ALNS

较SA和VNS的收敛速度慢,但两种对比算法陷入了局部最优,ALNS相比GA能以相近的速度收敛到更好的解;图7(c)中,ALNS较3种对比算法能收敛到更好的解,SA和VNS虽然收敛速度较快但易陷入局部最优. 图7(d)中,SA和VNS虽然收敛速度较快但易陷入局部最优,GA收敛速度较慢且得到的解也不如ALNS求得的解.

3.4 灵敏度分析

本节实验探究无人机载重能力和无人机续航能力这两个关键因素对实验结果的影响,算例选取P-n50-k7,配送时间的单位设置为小时.

1) 无人机载重能力.

实验将无人机的最大载重能力由1 kg增加至10 kg,4种算法的运行结果绘制在图8中. 由图8可见,随着无人机载重能力的变化,ALNS算法始终优于其他对比算法. 当无人机载重能力由1 kg增加至5 kg时,ALNS计算的配送时间下降最快:从4.28 h下降至2.78 h,而当无人机载重能力大于等于5 kg时逐渐趋于稳定. 实验结果表明无人机的载重能力会直接影响车辆与无人机的配送时间,当最大载重能力达到一

定限时,无人机的续航能力成为限制配送时间降低的重要因素.

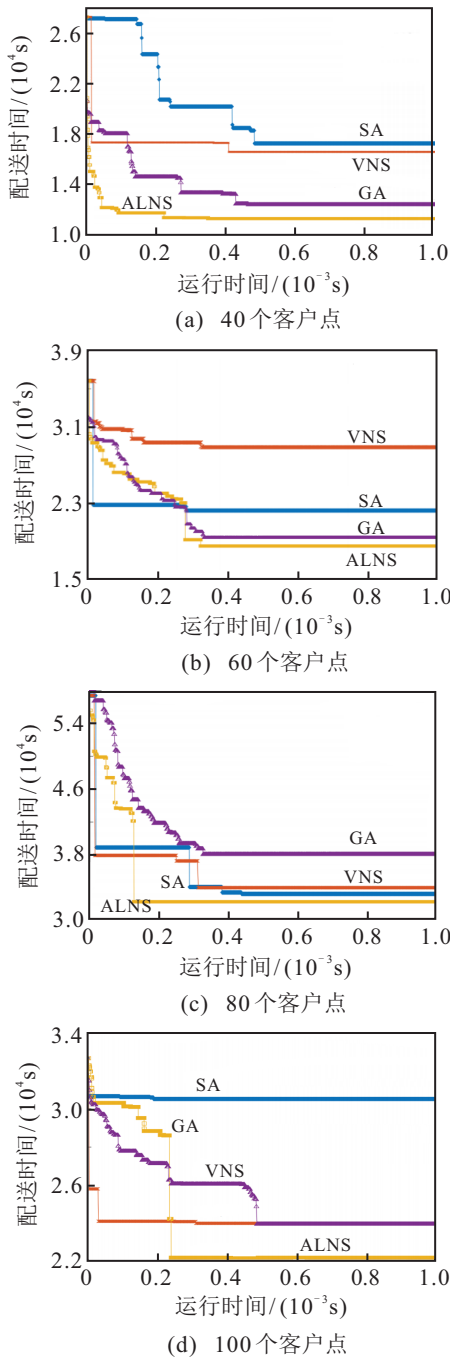


图7 不同规模算例的算法收敛曲线

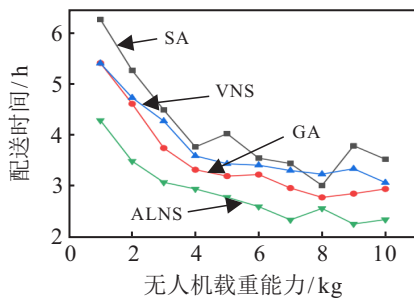


图8 不同无人机载重能力的实验结果

2) 无人机续航能力.

实验将无人机的最大电池能量从100J增加到

1100J,算法的运行结果绘制在图9中.由图9可见,随着无人机电池容量的增加,配送时间逐渐降低,当无人机电池能量由100J增加至500J时,ALNS优化效果明显改进,当电池能量大于等于500J时配送时间趋于稳定.该结果表明无人机的续航能力的提升是降低物流配送时间的有效因素,能够让无人机为车辆分担更多的配送任务,从而降低配送时间.

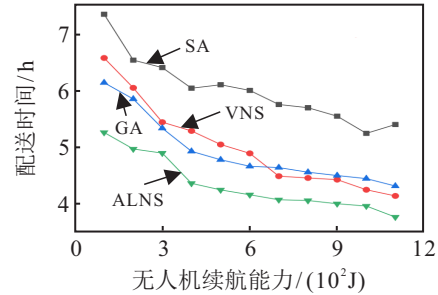


图9 不同无人机电池能量的实验结果

4 结论

本文对多辆车与多架无人机的协同配送问题进行研究,考虑了无人机一次飞行可配送多个包裹,且在配送过程中其能量消耗会随载荷重量的变化而变化.针对该问题,建立了以配送时间最短为目标的MIP,并提出了基于问题邻域知识的ALNS.实验结果表明,ALNS在小规模算例上求得的解与Gurobi求得的最优解相差甚微,相比SA、VNS、GA三种对比算法,ALNS的优化效果平均提升了23.8%、23.3%和5.7%,表明了ALNS能够针对不同实例以较少的时间求得更优的解.此外,灵敏度分析结果表明,无人机载重能力和无人机续航能力是影响配送时间的两个关键因素.

参考文献(References)

- [1] Murray C C, Chu A G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery[J]. *Transportation Research — Part C: Emerging Technologies*, 2015, 54: 86-109.
- [2] Scott J E, Scott C H. Drone delivery models for medical emergencies [J]. *Delivering Superior Health and Wellness Management with IoT and Analytics*, 2019, 28: 69-85.
- [3] Rupinder B, Benedict H, Wai S L, et al. Google's wing drones deliver essentials during coronavirus pandemic[EB/OL]. (2020-04-15)[2021-3-29]. <https://www.dezeen.com/2020/04/15/google-wing-drone-delivery-coronavirus-virginia/>.
- [4] Chen D W, Pan S L, Chen Q, et al. Vehicle routing problem of contactless joint distribution service during COVID-19 pandemic[J]. *Transportation Research Interdisciplinary Perspectives*, 2020, 8: 100233.
- [5] UPS has a delivery truck that can launch a drone[EB/OL]. (2017-02-21)[2021-3-29]. <https://www.ups.com>

- theverge.com/2017/2/21/14691062/ups-drone-delivery-truck-test-completed-video.
- [6] Agatz N, Bouman P, Schmidt M. Optimization approaches for the traveling salesman problem with drone[J]. *Transportation Science*, 2018, 52(4): 965-981.
- [7] Murray C C, Raj R. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones[J]. *Transportation Research—Part C: Emerging Technologies*, 2020, 110: 368-398.
- [8] Sacramento D, Pisinger D, Ropke S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones[J]. *Transportation Research—Part C: Emerging Technologies*, 2019, 102: 289-315.
- [9] Chen C, Demir E, Huang Y. An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots[J]. *European Journal of Operational Research*, 2021, 294(3): 1164-1180.
- [10] Kitjacharoenchai P, Ventresca M, Moshref-Javadi M, et al. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach[J]. *Computers & Industrial Engineering*, 2019, 129: 14-30.
- [11] Poikonen S, Golden B. Multi-visit drone routing problem[J]. *Computers & Operations Research*, 2020, 113: 104802.
- [12] Dorling K, Heinrichs J, Messier G G, et al. Vehicle routing problems for drone delivery[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017, 47(1): 70-85.
- [13] Moshref-Javadi M, Hemmati A, Winkenbach M. A comparative analysis of synchronized truck-and-drone delivery models[J]. *Computers & Industrial Engineering*, 2021, 162: 107648.
- [14] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows[J]. *Transportation Science*, 2006, 40(4): 455-472.
- [15] 孙棣华, 杨陈成, 廖孝勇, 等. 基于公交GPS数据的交叉口信号配时参数估计[J]. *控制与决策*, 2018, 33(4): 724-730.
(Sun D H, Yang C C, Liao X Y, et al. Signal timing estimation for intersections using bus GPS data[J]. *Control and Decision*, 2018, 33(4): 724-730.)
- [16] 李国明, 李军华. 基于混合禁忌搜索算法的随机车辆路径问题[J]. *控制与决策*, 2021, 36(9): 2161-2169.
(Li G M, Li J H. Stochastic vehicle routing problem based on hybrid tabu search algorithm[J]. *Control and Decision*, 2021, 36(9): 2161-2169.)
- [17] Ric. How much weight can delivery drones carry? -UnmannedCargo.org[EB/OL]. (2020-02-13) [2021-12-14]. <http://unmannedcargo.org/how-much-weight-can-delivery>.
- [18] Yamur E, Kesen S E. Multi-trip heterogeneous vehicle routing problem coordinated with production scheduling: Memetic algorithm and simulated annealing approaches[J]. *Computers & Industrial Engineering*, 2021, 161: 107649.
- [19] Hansen P, Mladenovi N. Variable neighborhood search[M]. Berlin: Springer, 2003: 145-184.
- [20] 何庆, 吴意乐, 徐同伟. 改进遗传模拟退火算法在TSP优化中的应用[J]. *控制与决策*, 2018, 33(2): 219-225.
(He Q, Wu Y L, Xu T W. Application of improved genetic simulated annealing algorithm in TSP optimization[J]. *Control and Decision*, 2018, 33(2): 219-225.)
- [21] Christofides N. The vehicle routing problem[J]. *Combinatorial Optimization*, 1979, 10(1): 315-338.
- [22] Augerat P, Naddef D, Belenguer J, et al. Computational results with a branch and cut code for the capacitated vehicle routing problem[R]. Grenoble: Université Joseph Fourier, 1995.
- [23] Liu Y, Luo Z H, Liu Z, et al. Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions[J]. *IEEE Access*, 2019, 7: 63504-63518.
- [24] Derrac J, García S, Molina D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. *Swarm and Evolutionary Computation*, 2011, 1(1): 3-18.

作者简介

伍国华(1986—), 男, 教授, 博士生导师, 从事智能优化与决策等研究, E-mail: guohuawu@csu.edu.cn;

毛妮(1996—), 女, 硕士生, 从事车辆与无人机路径规划的研究, E-mail: ni_mao@csu.edu.cn;

徐彬杰(1998—), 男, 博士生, 从事智能优化算法的研究, E-mail: binjie_xu@csu.edu.cn;

马中强(1991—), 男, 博士生, 从事仓储系统优化的研究, E-mail: mzq11302@163.com;

赵苛欣(1997—), 女, 硕士生, 从事车辆与无人机弧路由规划调度的研究, E-mail: kx_zhao@csu.edu.cn;

李志猛(1976—), 男, 教授, 博士生导师, 从事军事运筹学、资源运用优化等研究, E-mail: zmli@nudt.edu.cn.

(责任编辑: 郑晓蕾)