

时间依赖型带软时间窗的越库配送 绿色车辆路径问题

李雯, 倪静

(上海理工大学 管理学院, 上海 200093)

摘要: 针对越库配送模式下物流费用高、客户满意度低、碳减排压力大的问题, 提出了一种基于时变车速的跨时段道路行程时间的计算方法, 更加精准地度量成本与时间的关系。综合考虑软时间窗约束以及行驶速度对碳排放的影响, 构建了双目标模型, 同时优化系统总成本和等待延误时间这两个相互冲突的目标, 并提出了一种结合可变邻域搜索与非支配排序遗传算法II的混合启发式算法进行求解。数值实验表明, 该算法能在不同规模算例下有效求解模型, 算法综合性能更高。改进的时变车速模型在提升客户满意度方面优于静态网络。

关键词: 越库配送; 时间依赖; 绿色车辆路径问题; 可变邻域搜索; NSGA-II

中图分类号: TP 18 **文献标志码:** A

Time-dependent green vehicle routing problem for cross-docking with soft time windows

LI Wen, NI Jing

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In response to the problems of high logistics costs, low customer satisfaction and significant pressure for carbon reduction under vehicle routing problem for cross-docking, a method for calculating road travel time across time periods based on time-varying vehicle speeds was proposed, enabling more accurate assessments of cost and time. By comprehensively considering soft time window constraints and the impact of driving speed on carbon emissions, a bi-objective model was developed to simultaneously optimize the conflicting objectives of total system cost and waiting delay time. A hybrid heuristic algorithm combining variable neighborhood search (VNS) and non-dominated sorting genetic

收稿日期: 2024-11-19

基金项目: 国家自然科学基金资助项目(72174121); 国家社会科学基金资助项目(22BGL240)

第一作者: 李雯(2000—), 女, 硕士研究生。研究方向: 智能算法、路径优化。E-mail: liwen0021@163.com

通信作者: 倪静(1972—), 女, 副教授。研究方向: 智能算法、在线社会网络、企业信息化。E-mail: nijing501@126.com

引文格式: 李雯, 倪静. 时间依赖型带软时间窗的越库配送绿色车辆路径问题[J]. 上海理工大学学报, 2026, 48(1): 109-120.

Citation: LI Wen, NI Jing. Time-dependent green vehicle routing problem for cross-docking with soft time windows[J]. Journal of University of Shanghai for Science and Technology, 2026, 48(1): 109-120.

algorithm II (NSGA-II) was presented to solve the model. Numerical experiments show that the algorithm effectively solves the model across instances of varying scales and demonstrates high overall performance. The time-varying speed model enhances customer satisfaction compared to static networks.

Keywords: *cross-docking; time-dependent; green vehicle routing problem; variable neighborhood search; NSGA-II*

越库配送车辆路径问题 (vehicle routing problem with cross-docking, VRPCD) 是 VRP 的拓展问题。越库配送模式下, 货物无需长期存储在配送中心, 而是经过分拣、整合后直接转运至下游客户。相较于传统的配送策略, 越库配送在加速产品周转率、提高客户满意度、优化供应链效率等方面具有显著优势。然而, 不考虑现实情况往往导致此类问题的适应性差, 因此, 需要一种更合理的路径规划方案来解决 VRPCD 问题, 优化送取货路径, 在满足实际情况的同时实现降本增效。

Lee 等^[1]首次提出 VRPCD 模型, 探讨了拾取与交付的路径优化问题, 假定所有进站车辆同时到达越库中心并在同一时间处理, 使用禁忌搜索启发式算法优化固定成本和运输成本。为了解决同样的问题, Liao 等^[2]采用了一种改进的禁忌搜索算法进行求解, 在更短的时间内获得了更优的解决方案。Wen 等^[3]针对 VRPCD 模型作出改进, 放松了车辆同时进站的约束, 并增加了整合决策来确定车辆之间的依赖关系, 最大限度减少总运输时间。在此研究基础上, 学者们提出了不同的问题模型与求解算法, 丰富了越库配送模式在现实中的应用场景。Shahabi-Shahmiri 等^[4]研究了考虑易腐货物的 VRPCD 问题, 提出一种改进的约束增强方法进行求解, 提高了物流运输效率。范厚明等^[5]研究了多越库中心的 VRPCD 问题, 设计混合遗传算法, 有效降低了越库中心的运营成本。Cen 等^[6]研究了基于三维装载约束的 VRPCD 问题, 在大规模实例下获得高质量的解决方案。尽管越库策略已被证明在提高运输效率方面具有推进作用, 但越库中心需要足够的运输车队来运营, 有必要进一步研究运输过程中所产生的碳排放。相关 VRPCD 模型缺乏对环境因素的考量, 为此, 有学者围绕考虑碳减排的 VRPCD 问题进行了研究。Yaghoubi 等^[7]基于越库模式, 同时优化车辆运输成本和碳排放量, 实验表明, 该研究能指导企业实施利于可持续发展的物流决策。

Rajabzadeh 等^[8]最大限度地减少系统总成本和碳排放总量, 实验表明, 提高碳排放目标函数的重要性会影响车辆的选择以及货物的分配。Abad 等^[9]考虑越库中心的集成合并以及硬时间窗约束, 尽可能在降低系统总成本的同时缓解对环境的影响。然而, 对时间约束的硬性要求往往伴随着高额成本。因此, 在既考虑客户体验又考虑经济效益的情况下, 软时间窗约束显然更适用于实际情况。

关于越库配送的研究不仅需要加入软时间窗和碳排放这类实际需求, 也应考虑实际路网情况。由于道路网络容量的有限性, 拥堵已成为城市地区的日常现象, 特别是在早晚高峰时段, 车速会受到拥堵的影响而发生变化。为了模拟道路交通拥堵, Malandraki 等^[10]提出了时间依赖的车辆路径问题 (time-dependent vehicle routing problem, TDVRP), 将车辆在任意两点处的行驶时间视为出发时间的阶跃函数, 并设计了最邻近法和割平面算法, 解决小规模问题。但是早期的 TDVRP 的研究往往不符合 FIFO (first in first out) 原则, Ichoua 等^[11]引入了 FIFO 准则, 采用速度-时间相关函数来表示动态道路网络, 其中, 车辆的行驶速度是阶跃函数, 相应的行驶时间是分段连续函数。Fleischmann 等^[12]提出了另一种时变模型, 基于平滑阶跃函数对行程时间建模。Huang 等^[13]同时考虑路径柔性和时间依赖性, 构建不同路段的分段行程时间函数。Figliozzi 等^[14]提出了带时间窗的时间依赖型车辆路径问题 (time-dependent vehicle routing problem with time windows, TDVRPTW), 设计 4 种不同的时变路网, 并采用迭代路径构造改进算法对带有软、硬时间窗的 TDVRPTW 进行了测试。针对 TDVRPTW 问题, 国内外学者提出了相关扩展问题及求解方法。李楠等^[15]考虑城市交通拥堵情况, 提出一种混合离散灰狼算法, 实验证明了算法的有效性。Liu 等^[16]研究绿色 TDVRPTW 问题, 设计了一种大邻域搜索算法,

在大规模实例下得到高质量的解。何美玲等^[17]考虑碳排放因素, 设计一种混合蚁群算法进行求解, 有效提高物流企业的经济效益。Li等^[18]针对TDVRPTW问题, 设计了一种考虑时空距离的自适应大邻域搜索算法, 优化了总行程时间与顾客满意度。然而, 上述模型都是基于交付过程, 没有研究同时刻拾取和交付过程中的时变情况。

为使模型更贴近现实情况, 本文综合考虑碳排放、软时间窗和时变车速, 建立时间依赖型带软时间窗的越库配送绿色车辆路径问题(TDGVRPCDTW)模型。基于越库配送模式下对取送货车辆进行调度, 优化车辆路径, 实现系统总成本最低。车辆等待时间和延误时间最短, 更好地从企业角度衡量成本效益, 并保证一定的服务水平。最后, 提出一种改进的变邻域搜索与非

支配排序遗传算法Ⅱ相结合的启发式算法(VNS-NSGA-Ⅱ)来求解该问题。

1 数学模型

1.1 问题描述

越库配送网络中存在一个越库中心、若干个供应商与零售商, 如图1所示。由越库中心先安排取货车辆去各个供应商拾取货物, 集货完成回到越库中心。在越库中心不存在仓储环节, 直接根据零售商的订单需求对货物进行重新分拣分装, 再以最快的速度安排送货车辆到各个零售商进行配送任务, 最后返回至越库中心。供应商与零售商之间的供需关系已知, 即车辆从供应商取出的货物需送至指定的零售商处。

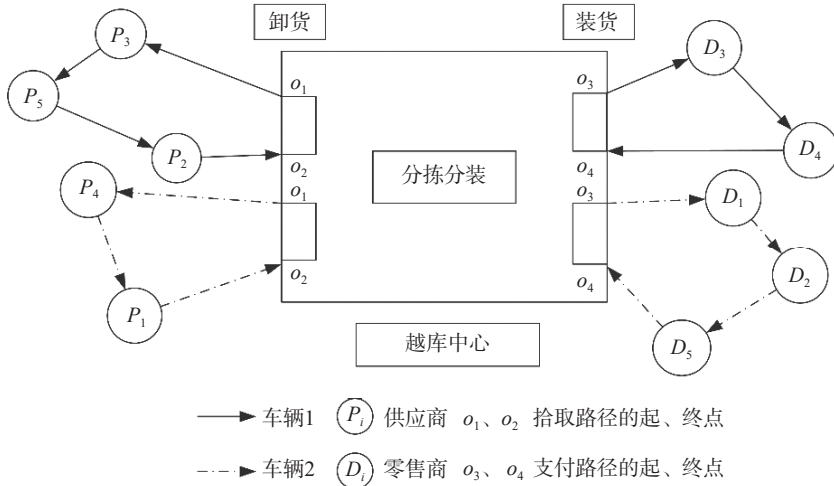


图1 越库网络

Fig.1 Cross-docking network

由于城市道路存在交通拥堵的情况, 因此, 车辆在不同的时间段以不同的速度行驶。每辆车从供应商拾取完成返回越库中心后, 可以立即开始卸货。在越库中心, 每辆车必须完成卸载任务之后才可以进行货物的重新装载。装卸货的持续时间包括一个固定的准备时间和装卸货订单所需的时间。

1.2 符号和决策变量

将TDGVRPCDTW定义为一个有向图 $G=(N,E)$ 。其中: 越库网络中所有点集合 $N=PUOUD$; 集合 E 表示网络中所有的可行弧; 拾取节点的集合表示为 $P=\{1,\dots,n\}$; 交付节点的集合表示为 $D=\{n+1,\dots,2n\}$; 车辆的集合表示为 $K=\{1,\dots,k\}$, 时间段的集合表示为 $H=\{1,\dots,h\}$ 。每个

请求 i 由节点对 $(i,i+n)$ 标识, 其中, i 是拾取节点, $i+n$ 是相对应的交付节点。越库中心有4个节点, 并由集合 $O=\{o_1,o_2,o_3,o_4\}$ 表示。 o_1 和 o_2 节点表示拾取路径的起始和终止位置, o_3 和 o_4 节点表示交付路径的起始和终止位置。 d_{ij} 表示弧 (i,j) 段的距离; θ 表示车辆单位距离的运输成本; φ 表示每辆车的固定成本; v 表示车辆的行驶速度; d_i 表示货物量; $[t_{ET,i},t_{LT,i}]$ 表示期望服务时间窗口; Q_m 表示车辆最大载重量; μ 表示碳排放单价; R_{ijk}^h 表示时间段 h 内的碳排放量; α 表示在越库中心装卸货的固定时间; β 表示每托盘货物的单位装/卸货时间。 t_{ijk} 为车辆 k 在 i,j 两点间的行驶时间; L_{ik} 为车辆 k 离开节点 i 的时间; e_k 为车辆 k 在越库中心完成卸载的时间; s_k 为车辆 k 在越库中心开始装载的时间;

f_i 为节点*i*的货物在越库中心完成卸载的时间； $t_{a,ik}$ 为车辆*k*到达节点*i*的时间； $t_{e,ik}$ 为车辆*k*到达节点*i*等待的时间； $t_{r,ik}$ 为车辆*k*到达节点*i*延迟的时间。决策变量： X_{ijk} 表示车辆*k*是否在弧(*i,j*)上行驶，是为1，否为0； x_{ijk}^h 表示在时间段*h*内车辆*k*是否在弧(*i,j*)上行驶，是为1，否为0； y_{ik} 表示车辆*k*是否访问节点*i*，是为1，否为0； u_{ik} 表示车辆*k*是否在越库中心卸载节点*i*的货物，是为1，否为0； r_{ik} 表示车辆*k*是否在越库中心装载节点*i*的货物，是为1，否为0； b_k 表示车辆*k*是否需要在越库中心卸载，是为1，否为0； w_k 表示车辆*k*是否需要在越库中心装载，是为1，否为0。

1.3 模型

旨在优化两个目标：最小化系统总成本以及等待和延迟时间。式(1)表示求取系统总成本最小，包括运输费用、固定费用和碳排放费用。式(2)表示求取车辆因偏离服务时间窗口而导致的等待和延迟时间最短。

$$\min f_1 = \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} X_{ijk} d_{ij} \vartheta + \sum_{j \in N} \sum_{k \in K} X_{o_1jk} \varphi + \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \sum_{h \in H} x_{ijk}^h R_{ijk}^h \mu \quad (1)$$

$$\min f_2 = \sum_{i \in N} \sum_{k \in K} (t_{e,ik} + t_{r,ik}) \quad (2)$$

1.3.1 碳排放成本

碳排放成本采用文献[19]中的MEET模型进行计算，碳排放估算函数 ε 和负荷修正函数 λ 为

$$\varepsilon^h(v) = \zeta_0 + \zeta_1 v + \zeta_2 v^2 + \zeta_3 v^3 + \frac{\zeta_4}{v} + \frac{\zeta_5}{v^2} + \frac{\zeta_6}{v^3} \quad (3)$$

$$\lambda^h(v, \delta) = \omega_0 + \omega_1 \delta + \omega_2 \delta^2 + \omega_3 \delta^3 + \omega_4 v + \omega_5 v^2 + \omega_6 v^3 + \frac{\omega_7}{v} \quad (4)$$

式中： ζ 、 ω 表示与车辆相关的修正系数，根据车辆实际载重量而制定； δ 表示车辆实际负载与车辆满载的比率。

根据碳排放估算函数和负荷修正函数可以进一步计算碳排放量，如下所示：

$$R_{ijk}^h = \varepsilon^h \lambda^h \frac{d_{ijk}^h}{1000} \quad (5)$$

1.3.2 具有时变车速的时间段内行驶时间

碳排放成本以及车辆到达节点的时间都直接依赖于车辆的行驶速度，而现实生活中车速会随时间不断发生变化，因此，考虑时变车速才能更加精准地度量成本和时间的计算。

为了克服分段行程时间函数中阶跃的缺点，

Ichoua等^[11]提出了一种将离散行程速度函数转化为连续行程时间函数的方法，确保车辆根据FIFO规则沿弧段行驶。因此，本文采用基于行驶速度与行驶时间的连续依赖模型。如图2所示，车辆的行驶速度是时间的分段函数，每个时段拥有相同的行驶速度，由此可以得到同一路段(长度为1)下不同出发时间的行驶时间函数。

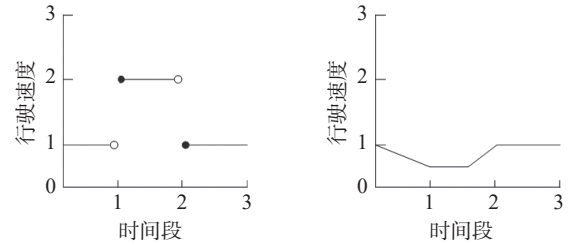


图2 行驶速度与行驶时间的关系

Fig.2 Relationship between travel speed and travel time

设*F*为时间段的长度； d_{ijk}^h 、 t_{ijk}^h 、 v_{ijk}^h 分别表示在时间段*h*内车辆*k*在弧(*i,j*)上的行驶距离、时间和速度； d_{ij}^h 表示车辆*k*在时间段*h*之后继续行驶通过整个弧(*i,j*)所需的剩余距离； r_{ijk}^h 表示在时间段*h*内车辆*k*沿弧(*i,j*)行驶的剩余时间。车辆*k*在弧(*i,j*)上的行驶时间 t_{ijk} 通过以下步骤进行计算：

步骤1 计算初始时间段的行驶时间。 $d_{ijk}^h = v_{ijk}^h r_{ijk}^h$ ， $r_{ijk}^h = Fh - L_{ik}$ ，判断当前时间段车辆能否从节点*i*行驶到节点*j*。如果 $d_{ijk}^h \geq d_{ij}$ ，则 $t_{ijk}^h = d_{ij} / v_{ijk}^h$ ， $t_{ijk} = t_{ijk}^h$ ，结束计算；如果 $d_{ijk}^h < d_{ij}$ ，表明车辆需要跨时间段行驶，则计算剩余距离 $d_{ij}^h = d_{ij} - d_{ijk}^h$ ， $t_{ijk}^h = r_{ijk}^h$ ，继续计算下一个时间段。

步骤2 计算其他时间段的行驶时间。 $d_{ijk}^h = v_{ijk}^h F$ ，判断当前时间段车辆是否需要继续跨时段行驶，如果 $d_{ijk}^h < d_{ij}^h$ ，表明当前时间段车辆还不能到达目的地，则 $d_{ijk}^{h+1} = d_{ij}^h - d_{ijk}^h$ ， $t_{ijk}^h = F$ ，再继续步骤2的方法跨时间段计算，直到 $d_{ijk}^h > d_{ij}^h$ ，

$t_{ijk}^h = d_{ij}^h / v_{ijk}^h$ ， $t_{ijk} = \sum_{h \in H} t_{ijk}^h$ ，结束计算。

1.3.3 约束条件

$$\sum_{j:(i,j) \in E} \sum_{k \in K} X_{ijk} = 1, \forall i \in P \cup U \quad (6)$$

$$\sum_{i \in P} \sum_{j:(i,j) \in E} d_i X_{ijk} \leq Q_m, \forall k \in K \quad (7)$$

$$\sum_{i \in D} \sum_{j:(i,j) \in E} d_i X_{ijk} \leq Q_m, \forall k \in K \quad (8)$$

$$\sum_{j:(i,j) \in E} X_{ijk} = 1, \forall i \in \{o_1, o_3\}, \forall k \in K \quad (9)$$

$$\sum_{j:(i,j) \in E} X_{ijk} = 1, \forall i \in \{o_2, o_4\}, \forall k \in K \quad (10)$$

$$\sum_{i \in P \cup \{o_1\}} X_{ijk} = y_{ik}, \forall j \in P, k \in K \quad (11)$$

$$\sum_{i \in D \cup \{o_3\}} X_{ijk} = y_{ik}, \forall j \in D, k \in K \quad (12)$$

$$\sum_{j:(j,i) \in E} X_{jik} = \sum_{j:(i,j) \in E} X_{ijk}, \forall i \in P \cup U \quad (13)$$

$$t_{a,jk} \geq t_{a,ik} + t_{ijk} + \beta d_i - M(1 - X_{ijk}), \\ \forall (i, j) \in E, k \in K \quad (14)$$

$$L_{ik} = t_{a,ik} + \beta d_i, \forall k \in K \quad (15)$$

$$t_{e,ik} \geq t_{ET,i} y_{ik} - t_{a,ik}, \forall i \in N, k \in K \quad (16)$$

$$t_{r,ik} \geq t_{a,ik} - t_{LT,i} y_{ik}, \forall i \in N, k \in K \quad (17)$$

$$t_{ET,i} y_{ik} \leq t_{a,ik} \leq t_{LT,i} y_{ik}, \forall i \in N, k \in K \quad (18)$$

$$t_{ijk} = \sum_{h \in H} X_{ijk} t_{ijk}^h, \forall (i, j) \in E, k \in K \quad (19)$$

$$d_{ijk} = \sum_{h \in H} X_{ijk} d_{ijk}^h, \forall (i, j) \in E, k \in K \quad (20)$$

$$u_{ik} - r_{ik} = \sum_{j \in P \cup \{o_2\}} X_{ijk} - \sum_{j \in D \cup \{o_4\}} X_{(i+n)jk}, \\ \forall k \in K, i \in P \quad (21)$$

$$u_{ik} + r_{ik} \leq 1, \forall k \in K, i \in P \quad (22)$$

$$\frac{1}{M} \sum_{i \in P} u_{ik} \leq b_k \leq \sum_{i \in P} u_{ik}, \forall k \in K \quad (23)$$

$$e_k = t_{a,o_2k} + \alpha b_k + \beta \sum_{i \in P} d_i u_{ik}, \forall k \in K \quad (24)$$

$$s_k \geq e_k, \forall k \in K \quad (25)$$

$$s_k \geq f_i - \Phi(1 - r_{ik}), \forall i \in P, \forall k \in K \quad (26)$$

$$f_i \geq e_k - \Phi(1 - u_{ik}), \forall i \in P, \forall k \in K \quad (27)$$

$$\frac{1}{\Phi} \sum_{i \in P} r_{ik} \leq w_k \leq \sum_{i \in P} r_{ik}, \forall k \in K \quad (28)$$

$$t_{a,o_3k} = s_k + \alpha w_k + \beta \sum_{i \in P} d_i r_{ik}, \forall k \in K \quad (29)$$

$$t_{ET,o} \leq t_{a,o_3k} + t_{o_3ik} + \\ \sum_{j:(i,j) \in E} t_{ijk} + t_{j o_4k} \leq t_{LT,o}, \forall k \in K \quad (30)$$

$$X_{ijk}, x_{ijk}^h, y_{ik}, u_{ik}, r_{ik}, b_k, w_k \in \{0, 1\} \quad (31)$$

$$t_{ijk}, d_{ijk}, d_{ij}, t_{ijk}^h, d_{ijk}^h, v_{ijk}, d_{ij}^h, e_k, \\ s_k, t_{ijk}, t_{a,ik}, t_{e,ik}, t_{r,ik} \geq 0 \quad (32)$$

约束(6)表示每个供应商与零售商只能被一辆车服务一次。约束(7)和(8)表示对于每辆车、拾取路线和交付路线上的负载不能超过车辆最大负载量。约束(9)和(10)规定每辆车必须从越库中心出发, 然后返回越库中心。约束(11)和(12)确保车辆到达供应商或零售商节点必须向前移动到下一个供应商或零售商节点。约束(13)表示流量守恒约束。约束(14)计算车辆 k 到达节点 j 的时间。约束(15)计算车辆 k 离开节点 i 的时间。约束(16)和(17)表示当车辆早于期望时间到达或者车辆晚于期望时间到达时所发生的提前和延迟。约束(18)表示每个车辆在允许时间窗口内到达节点需要满足的约束。约束(19)表示每个时间段内路段行驶时间与车辆行驶时间之间的关系。约束(20)表示每个时间段内路段距离与车辆行驶距离之间的关系。约束(21)和(22)表示车辆 k 是否在越库中心卸载以及装载供应商 i 的货物。考虑了以下3种情况: a. 如果车辆 k 在节点 i 处拾取但不交付给节点 $i+n$, 则需要在越库中心卸载节点 i 的货物; b. 如果车辆 k 不在节点 i 处拾取但交付给节点 $i+n$, 则需要在越库中心重新装载节点 i 的货物; c. 如果车辆既在节点 i 处拾取且要交付给节点 $i+n$, 则不需要装载或卸载节点 i 的货物。约束(23)表示车辆是否需要要在越库中心卸载。约束(24)表示车辆在越库中心完成卸载的时间等于车辆到达越库中心的时间加上车辆准备时间以及卸货时间, 其中卸货时间等于货物重量乘以单位重量卸货时间。约束(25)~(27)确保车辆必须在完成全部卸载之后才能开始重新装载, Φ 表示一个很大的数。约束(28)表示车辆是否需要在越库中心装载。约束(29)表示车辆在越库中心结束装载的时间等于车辆开始装货的时间加上车辆准备时间以及实际装货时间。约束(30)表示返回越库中心的车辆应受限于越库中心时间窗口的约束。约束(31)和(32)定义变量的范围。

2 算法设计

由于 TDGVRPCDTW 是 NP 难问题, 很难在有限的时间里找到高质量的解, 采用元启发式方法来解决。NSGA-II 已被验证能有效解决越库配送车辆路径问题^[20], 具有解集收敛性好、全局搜索能力强的优点。可变邻域搜索(VNS)是一种基于局部搜索的方法, 能够系统地改变邻域结构,

在局部区域进行精细搜索，增强解集的多样性，避免陷入局部最优^[21]。在NSGA-II算法的基础上，利用VNS的局部优化，能够进一步改善生成解的质量，使得最终得到的解集更加接近真实的Pareto前沿。因此，本文提出了一种变邻域搜索结合非支配排序遗传算法II的混合元启发式算法(VNS-NSGA-II)。图3给出了VNS-NSGA-II算法的详细流程。

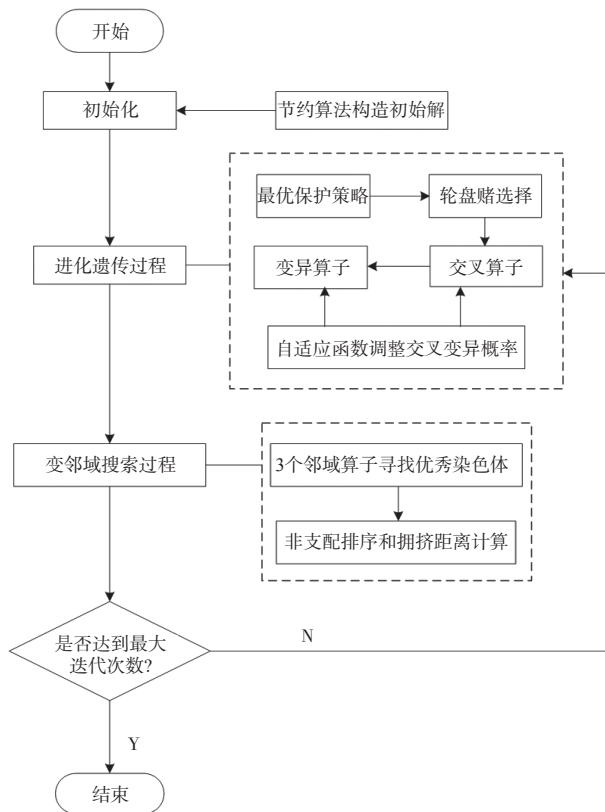


图3 VNS-NSGA-II 算法流程

Fig.3 VNS-NSGA-II algorithm flow

2.1 编码与解码

本研究采用随机方法生成初始种群，通过整数编码的方式随机生成多条长度为节点数的染色体。所提出的问题解决方案包括拾取和交付过程两个部分，染色体的编码与解码过程如图4所示。以8个供应商-零售商节点对为例，第一行表示供应商节点，第二行表示零售商节点。在将染色体解码为取送货路径时，根据车辆的载重量和时间相关的约束对染色体进行分段。假设在供应商7位置处不能满足约束条件，则在供应商7前插入越库中心0，继续为下一个节点重新安排车辆。重复上述过程，直至所有染色体完成路径划分，最终形成3条拾取路径和3条交付路径。

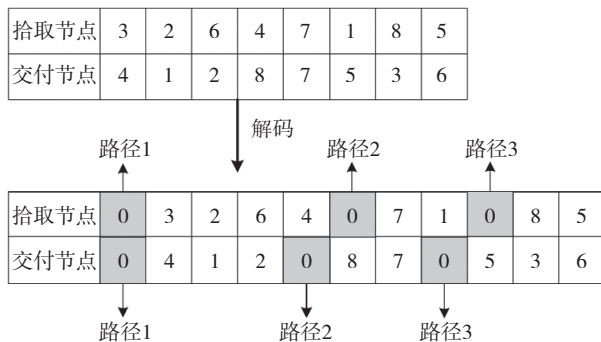


图4 编码与解码

Fig.4 Encoding and decoding process

2.2 初始解生成

初始解生成对于变邻域搜索的求解质量有较大影响。高质量的初始解可以提高算法的性能，帮助算法寻找到更优的解。因此，本文采用节约算法构造初始解提高初始种群的质量。针对TDGVRPCDTW模型中供应商与零售商一一对应的供需关系，首先为每个供应商-零售商节点对分配一辆空车，也就是同时构造取货与送货路径，随后每次迭代过程中将节约值最大的2条路径进行合并，直至不能减少目标函数值时终止迭代。初始解生成步骤具体如下：

- a. 为每个供应商-零售商节点对分配一辆空车，即每个供应商-零售商节点对由同一辆车服务。
- b. 将任意两条路径进行合并，计算节约值，定义为 $S = (d_{io2} + d_{o1j}) + (d_{i'o4} + d_{o3j'}) - d_{ij} - d_{i'j'}$ ，其中： i, i' 表示第一条路径上供应商-客户对； j, j' 表示第二条路径上供应商-节点对。
- c. 降序排列节约值表，在不违反车辆容量限制的条件下选取节约值最大的两条路径进行合并。
- d. 重复步骤b~c，直至合并任意两条路径不能达到节约目标函数值时结束算法。

2.3 交叉和变异

交叉过程是指在父代染色体上随机选择两个交叉点，让两条父代染色体中交叉点的前后片段进行交叉，生成新的子代染色体。交叉完成后保留两个交叉部分中的染色体片段，并将剩余位置中的重复片段删除。通过交换路径中客户的访问顺序，寻找更加经济的路线，同时减少车辆不必要的等待或者延误时间，从而提高客户满意度。变异过程是指随机选择两个突变点，将这两个节点进行交换以生成新的子代染色体，增加路径方案的灵活性，更好地满足客户的时间窗约束，也有助于寻找到潜在的低成本方案。

由于交叉概率和变异概率是决定算法寻优能力的关键参数, 交叉概率和变异概率过高或者过低都会影响算法寻找优秀的个体, 而大部分越库配送车辆路径问题中交叉和变异概率都被设置为固定的值。本文采用文献 [18] 提出的自适应函数动态调整交叉与变异概率, 最大限度搜寻利于降低成本且提高客户满意度的解集。自适应交叉和变异概率的计算过程如式 (33) 和 (34) 所示。

$$p_c = \begin{cases} p_{c1} - \frac{(p_{c1} - p_{c2})(f_1 - f_{\text{avg}})}{(f_{\text{max}} - f_{\text{avg}}) \left[1 + \exp\left(\frac{-MS_{\text{non}}}{M_{\text{max}} S_{\text{size}}}\right) \right]}, & f_1 \geq f_{\text{avg}} \\ p_{c1} \left[1 + \exp\left(\frac{-MS_{\text{non}}}{M_{\text{max}} S_{\text{size}}}\right) \right], & f_1 < f_{\text{avg}} \end{cases} \quad (33)$$

$$p_m = \begin{cases} p_{m1} - \frac{(p_{m1} - p_{m2})(f_{\text{max}} - f_1)}{(f_{\text{max}} - f_{\text{avg}}) \left[1 + \exp\left(\frac{MS_{\text{non}}}{M_{\text{max}} S_{\text{size}}}\right) \right]}, & f_1 \geq f_{\text{avg}} \\ p_{m1} \left[1 + \exp\left(\frac{MS_{\text{non}}}{M_{\text{max}} S_{\text{size}}}\right) \right], & f_1 < f_{\text{avg}} \end{cases} \quad (34)$$

式中: p_c 和 p_m 分别表示自适应交叉概率和变异概率; p_{c1} 、 p_{c2} 、 p_{m1} 和 p_{m2} 为自适应系数, $p_{c1} > p_{c2}$, $p_{m1} > p_{m2}$; f_1 表示染色体中较大的适应度值; f_{avg} 表示种群中适应度函数值的平均值; f_{max} 表示种群中最大的适应度函数值; M 为迭代次数; M_{max} 是最大迭代次数; S_{size} 是种群规模大小; S_{non} 是染色体没有发生改变的数量。

2.4 可变邻域搜索

为了避免陷入局部最优, 提高解集的质量, 本文在 NSGA-II 算法的基础上加入 VNS 策略, 通过不同的邻域搜索算子, 系统地探索更大的邻域空间, 逐步更新优秀的解集。该策略在找到第一个比当前解更好的解时, 则停止当前搜索并进行下一步迭代。本文设定 3 种邻域算子, 分别是 Intra-shift、Inter-shift、2_opt。Intra-shift 表示将某个供应商或零售商节点从当前位置移除, 并插入到当前路径的另一可行位置。Inter-shift 表示将某个供应商或零售商节点从当前位置移除, 并插入到另一路径的可行位置。根据 TDGVRPCDTW 中先取货后送货的问题特征, 设计了 2-opt 算子。区别于反转两点间路径服务顺序的思路, 本文随机选择给定路径中的两个供应商或零售商节点。若选择的两个节点分别是取货点和送货点, 则分别反转取货与送货路径, 以此确保供应商和零售商的先后

服务顺序。

通过重新组织路径中的节点, 可以优化路径方案, 减少行驶距离和车辆的使用数量, 从而降低总成本。利用不同的邻域结构可以更好地满足客户的时间要求, 减少违反时间窗口的情况, 提高客户的满意度。调整路径中客户顺序的灵活性使得算法也能够更好地应对突发情况, 例如交通堵塞等, 从而降低应对异常情况的成本, 提高客户的满意度。

3 仿真实验

3.1 数据描述和参数设置

由于目前无基准算例可用于求解 TDGVRP-CDTW 问题, 因此, 实验基于文献 [3] 公开的 VRPCD 标准算例进行修改以生成不同规模的 TDGVRPCDTW 算例。其中: 供应商、零售商和越库中心地理位置信息、供应商的供应量、零售商的需求量、时间窗范围数据均提取自 VRPCD 数据集。时变车速以及碳排放计算的相关参数则是参照文献 [16]: 越库中心的时刻 0 表示早上 7 点, 交通拥堵时段为 7:00—9:00、17:00—19:00, 拥堵车速为 20 km/h。剩余时间段 h 具有 3 个时变速度, 分别设为 54、72、42 km/h。碳排放模型的相关系数如下: $\zeta_0=110$, $\zeta_1=0$, $\zeta_2=0$, $\zeta_3=0.000375$, $\zeta_4=8702$, $\zeta_5=0$, $\zeta_6=0$, $\omega_0=1.27$, $\omega_1=0.0614$, $\omega_2=0$, $\omega_3=-0.0011$, $\omega_4=-0.00235$, $\omega_5=0$, $\omega_6=0$, $\omega_7=-1.33$ 。碳交易价格 $\mu=0.0528$ 元/kg, 单位距离运输成本 $\theta=1.3$, 使用车辆的固定成本 $\varphi=20$ 。

算法程序均采用 Matlab R2021 编程, 在主频 1.19 GHz 的 Intel Core i5-1035 处理器及 8GB RAM 的 Windows 10 电脑操作系统上实现。具体算法参数设置如下: 种群规模 $S_{\text{size}}=200$, 最大迭代次数 M_{max} 为 200~450。由于算例规模不同, 越大规模的算例需要更大的迭代次数以接近真实 Pareto 解, 因此, 为各算例设置不同的迭代次数。小规模设置为 200, 中规模设置为 300, 大规模设置为 450。交叉概率 $p_{c1}=[0.5, 0.8]$, $p_{c2}=[0.4, 0.6]$, 变异概率 $p_{m1}=[0.01, 0.03]$, $p_{m2}=[0.002, 0.01]$ 。

由于问题涉及双目标优化, 采用算法评价指标反转世代距离 (IGD) 和超容量 (HV) 来评估算法的性能。IGD 指标用来度量多目标优化问题非支配解的收敛性和多样性。IGD 值越小, 说明算法

综合性能越好, 本文用 ξ_{IGD} 表示。对于真实的最优 Pareto 前沿解中的每个解 σ_{ture} , 找到与其最近的 PF 中的非支配解 σ_{solu} , 计算两者的距离并取平均值, 求解过程为

$$\xi_{IGD}(\sigma_{solu}, \sigma_{ture}) = \frac{\sum_{x \in \sigma^*, y \in \sigma_{solu}} \max \text{dis}(x, y)}{|\sigma_{ture}|} \quad (35)$$

真实的 Pareto 前沿解 σ_{ture} 是未知的, 因此, 使用从所有算法结果中获得的 Pareto 最优解集 σ^* 近似真实的 Pareto 前沿解 σ_{ture} 。

HV 指标是非支配解集与参考点之间组成的超立方体的体积。从收敛性和多样性的角度来看, 具有较高 HV 值的集合表示逼近 Pareto 前沿的较好的非支配解集合, 并且 HV 值越高, 说明算法的收敛性和多样性更好。本文用 ξ_{HV} 表示 HV 值, 求解过程为

$$\xi_{HV}(\sigma_{solu}, r) = \frac{\text{volume}(\cup_{s \in \sigma} [f_1(s), f_{\max_1}] [f_2(s), f_{\max_2}])}{\text{volume}(\cup_{s \in \sigma} [f_1(s), f_{\max_1}] [f_2(s), f_{\max_2}])} \quad (36)$$

式中: 参考点 $r = (f_{\max_1}, f_{\max_2})$ 由集合 σ_{solu} 中的任何元素支配, 每个元素都是目标函数在已知最优解上的最大值; $[f_i, f_{\max_i}]$ 表示由集合 σ_{solu} 中的第 i 个解和参考点形成的超立方体。

3.2 测试算法的性能

3.2.1 TDGVRPCDTW 算例对比结果

本文将 VNS-NSGA-II 算法与 NSGA-II 算法、MOEA/D 算法以及 MOLNS 算法进行比较, 使用各种不同规模的数据集来验证本文提供的 VNS-NSGA-II 算法的效率。为了更直观地表现出算法的性能, 对各算法运行相同时间, 算例-50 运行 10 min, 算例-100 运行 25 min, 算例-150 运行 60 min, 算例-200 运行 90 min, 每组算例独立运行 10 次, 取平均值。通过 IGD 值和 HV 值进行对比分析, 最优值加粗显示, 结果见表 1。

由表 1 可知, VNS-NSGA-II 算法在不同规模算例下基本都能取得最小的 IGD 值和最大的 HV 值, 说明了该算法的综合性能较高。这表明该

表 1 算法性能比较

Tab.1 Comparison of algorithm performance

算例	NSGA-II		MOEA/D		MOLNS		VNS-NSGA-II	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV
a-50	0.1233	0.8214	0.0146	0.8811	0.0179	0.8954	0.0051	0.9011
b-50	0.0621	0.7039	0.0221	0.9236	0.0211	0.9294	0.0106	0.9355
c-50	0.1413	0.8207	0.0104	0.8338	0.0142	0.8609	0.0127	0.9029
d-50	0.1080	0.8745	0.0069	0.8977	0.0039	0.9147	0.0041	0.9078
e-50	0.0871	0.8265	0.0117	0.8615	0.0121	0.8654	0.0125	0.8691
平均值-50	0.1044	0.8094	0.0131	0.8795	0.0138	0.8932	0.0090	0.9033
a-100	0.0751	0.8144	0.0256	0.8276	0.0152	0.8397	0.0188	0.8490
b-100	0.1034	0.7231	0.0144	0.8356	0.0127	0.8422	0.0119	0.8461
c-100	0.0865	0.8151	0.0213	0.8203	0.0189	0.8325	0.0141	0.9047
d-100	0.2305	0.7097	0.0273	0.8143	0.0255	0.8733	0.0216	0.9137
e-100	0.1306	0.6123	0.0217	0.8022	0.0117	0.8125	0.0148	0.8257
平均值-100	0.1252	0.7349	0.0221	0.8200	0.0168	0.8400	0.0162	0.8678
a-150	0.2761	0.5133	0.0378	0.7423	0.0275	0.8223	0.0254	0.8509
b-150	0.1826	0.7233	0.0393	0.7361	0.0314	0.8229	0.0268	0.8447
c-150	1.0149	0.6038	0.0318	0.8164	0.0258	0.8455	0.0219	0.9101
d-150	0.4233	0.7312	0.0376	0.7312	0.0328	0.7620	0.0284	0.8137
e-150	0.5108	0.6019	0.0296	0.7243	0.0277	0.7652	0.0260	0.8312
平均值-150	0.4815	0.6347	0.0352	0.7501	0.0290	0.8036	0.0257	0.8501
a-200	0.6438	0.4144	0.0469	0.6075	0.0386	0.6809	0.0344	0.7115
b-200	0.4844	0.5231	0.0577	0.7093	0.0479	0.7650	0.0412	0.8229
c-200	0.7712	0.4379	0.0535	0.8274	0.0408	0.8275	0.0296	0.8366
e-200	0.6713	0.2878	0.0562	0.7054	0.0426	0.7158	0.0310	0.7369
e-200	0.4677	0.5416	0.0557	0.6098	0.0455	0.6553	0.0397	0.7013
平均值-200	0.6077	0.4410	0.0540	0.6919	0.0431	0.7289	0.0352	0.7618

算法相较于 NSGA-II 算法在解的收敛性和多样性上具有显著优势, 也充分说明了可变邻域搜索纳入 NSGA-II 框架的优势。

进一步观察到 4 种比较算法的 IGD 和 HV 值的共同变化趋势, 随着节点对数量的增加, IGD 值有所增加, HV 值有所下降。这可能是由于随着节点对数量的增加, 问题复杂性增加, 算法寻优能力也会受到限制。不过综合来看, VNS-NSGA-II 可以在合理的计算时间内找到更高质量的 Pareto 前沿解。

3.2.2 Pareto 最优前沿解比较

为进一步直观地反映算法性能, 本文根据 4 种算法运行得到的目标函数最优值绘制了 Pareto 最优前沿解对比图。图 5 显示了 4 种算法在 4 种不同规模算例下获得的 Pareto 最优前沿解, 图中的每个点对应于由相应算法确定的 Pareto 最优前沿中的解。

如图 5 所示, 本文中描述的 VNS-NSGA-II 算法提供的解决方案在各种规模下都显著优于其他算法, 这意味着通过 VNS-NSGA-II 获得的 Pareto

最优解集具有更好的质量。

表 2 显示了 4 种算法得到的 Pareto 前沿中单个优化目标的最优值。从表中可以看出 VNS-NSGA-II 算法在所有规模的数据集上都实现了两个优化目标的最优结果。与 NSGA-II 算法相比, 总成本降低了 17.15%, 等待延误时间缩短了 0.75 min, 在成本上有明显改进。与 MOEA/D 算法相比, 总成本降低了 9.39%, 等待延误时间缩短了 11.75 min, 在等待延误时间方面有明显改进。与 MOLNS 算法相比, 总成本降低了 5.62%, 等待延误时间缩短了 1.25 min。上述结果反映出 VNS-NSGA-II 具有优越性, 算法的综合性能更好。

3.3 时变车速对结果的影响

在以往的 VRPCD 相关研究中, 部分研究者忽略了车速的时间依赖性, 相应模型的求解结果与实际结果可能存在一定偏差。为了分析不同车辆行驶速度对目标的影响, 在不同规模算例下比较静态道路网络与动态道路网络的目标函数值。其中, 静态道路网络下假定车辆行驶速度保持不变, 为 50 km/h, 时变道路网络下的车辆行驶速度

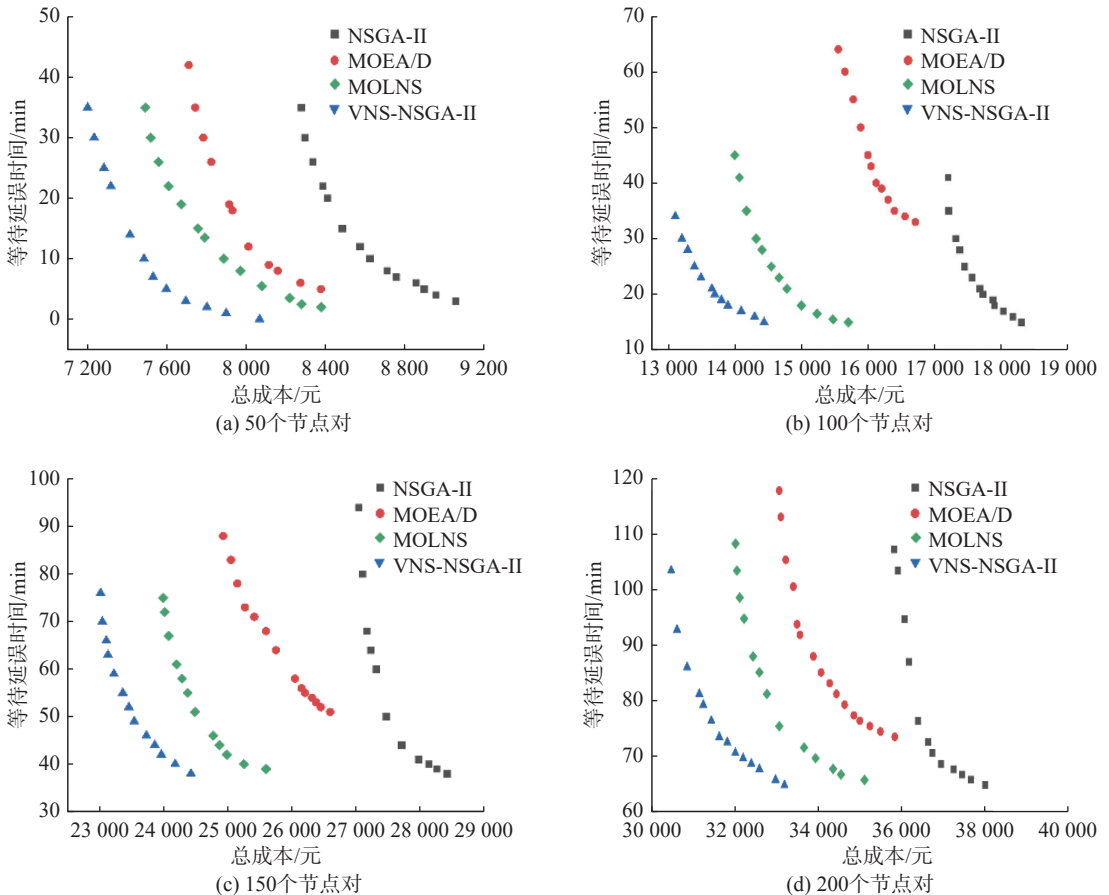


图 5 不同数据规模的 Pareto 最优前沿解比较

Fig.5 Comparison of Pareto frontier solution at different scales

表2 优化目标的比较结果

Tab.2 The comparison results of optimization objectives

算例	VNS-NSGA-II		NSGA-II		MOEA/D		MOLNS		与NSGA-II比较		与MOEA/D比较		与MOLNS比较	
	f_1 /元	f_2 /min	f_1 /元	f_2 /min	f_1 /元	f_2 /min	f_1 /元	f_2 /min	f_1 改进程度/%	f_2 /min	f_1 改进程度/%	f_2 /min	f_1 改进程度/%	f_2 /min
50	7200.21	0	8137.59	3	7686.38	5	7518.33	4	11.52	3	6.33	5	4.23	4
100	13007.26	15	17405.69	15	15565.04	33	14070.23	15	25.27	0	16.43	18	7.55	0
150	23026.74	38	27045.67	38	24935.38	51	24213.82	39	14.86	0	7.65	13	4.90	1
200	30499.45	65	36043.44	65	33189.03	74	32325.17	65	15.38	0	8.10	11	5.65	0
平均	18433.42	29.5	22248.09	30.25	20343.96	40.75	19531.89	30.75	17.15	0.75	9.39	11.75	5.62	1.25

为本文描述的数据。实验结果如表3所示，数据均是利用VNS-NSGA-II算法得到的，选取50~200不同规模的算例进行实验。

从表3可以看出，在不同规模的算例下，考虑时变车速相较于恒定车速对目标函数值会有不同的改进程度。对于50个节点对的小规模算例，时变网络下的平均成本比静态网络多出0.6%，等待延误时间平均改进22 min。对于100个节点对的中等规模算例，时变网络下的平均成本比静态网络多出0.7%，等待延误时间平均改进29 min。对于150个节点对的中等规模算例，时变网络下

的平均成本比静态网络多出0.4%，等待延误时间平均改进45 min。这对于200个节点对的大规模算例，时变网络下的平均成本比静态网络多出0.3%，等待延误时间平均改进67 min。这表明本文考虑车速的时间依赖性会造成一定的成本增量，这可能是由于拥堵时段车速降低，车辆使用数量增加，进而导致总成本上升。而本文的模型对改进车辆等待延误的情况有积极作用，尤其是在中大规模的算例下，改进程度更高。尽管考虑时变车速总成本会增加，但是准时率增加，客户满意度的提升更为显著，有利于企业的长远发

表3 时变车速对结果的影响

Tab.3 Influence of time-dependent speed on results

算例	静态网络		TDGVRPCDTW		f_1 改进程度/%	f_2 改进程度/min
	f_1 /元	f_2 /min	f_1 /元	f_2 /min		
a-50	7467.63	14	7511.75	5	0.59	9
b-50	7453.32	19	7476.46	8	0.31	11
c-50	7516.91	17	7589.68	8	0.97	9
平均值-50	7479.29	17	7525.96	7	0.62	10
a-100	13294.55	46	13374.32	21	0.60	25
b-100	13379.54	41	13463.21	22	0.63	19
c-100	13596.83	39	13714.79	18	0.87	21
平均值-100	13423.64	42	13517.44	20	0.70	22
a-150	22722.61	83	22834.68	41	0.49	42
b-150	22379.40	87	22465.17	42	0.38	45
c-150	23027.34	91	23156.46	47	0.56	44
平均值-150	22709.78	87	22818.77	43	0.48	44
a-200	30336.22	103	30378.25	58	0.14	45
b-200	30175.87	126	30284.64	50	0.36	76
c-200	30263.29	129	30462.92	64	0.66	65
平均值-200	30258.46	119	30375.27	57	0.39	62

展。因此, 本文提出的模型更加贴近实际运输情况, 具有重要的研究意义。

3.4 碳交易价格敏感性分析

碳交易价格的变化会直接导致碳排放量发生变化, 而碳排放量的变化会造成总成本波动, 进而影响到运输车辆的路径安排。本文设定的碳交易价格为 0.0528 元/kg, 现将碳交易价格设定为 0.0128、0.0328、0.0828、0.1028 元/kg。选取节点规模为 100 的算例进行实验, 分别运行 5 次, 取其平均值, 结果如表 4 所示。

表 4 不同碳交易价格对总成本与碳排放的影响

Tab.4 Influence of time-dependent speed on results

碳交易价格/(元·kg ⁻¹)	碳排放量/元	碳排放成本/元	总成本/元
0.0128	4354.17	55.73	13014.23
0.0328	4287.65	140.63	13175.49
0.0528	4223.16	222.98	13308.72
0.0828	4134.81	342.36	13539.06
0.1028	4059.75	417.34	13764.24

从表 4 可以看出, 碳交易价格与总成本成正比, 与碳排放量成反比。随着碳交易价格的增大, 碳排放成本增加的幅度远远超过了碳排放量的减少幅度。研究表明, 想要降低碳排放量并不能仅仅提高碳交易价格, 而是需要适当调整碳交易价格, 才能帮助企业在降低碳排放量的同时提高整体经济效益。通过设定不同范围的碳交易价格, 发现灵活的碳交易价格对企业减少碳排放有积极影响。

4 结束语

本文研究时间依赖型带软时间窗的越库配送绿色车辆路径问题(TDGVRPCDTW), 建立了一个双目标混合整数线性规划(MILP)模型, 该模型最小化物流总成本(包括运输成本、固定成本和碳排放成本)和等待延误时间。提出了一种基于时变车速的跨时段道路行程时间的计算方法, 更加精准地度量成本与时间的计算。为了解决这个问题, 提出了一种可变邻域搜索结合非支配排序遗传算法 II(VNS-NSGA-II)的混合启发式算法求解模型, 利用节约算法生成初始解。在遗传进化阶段, 采用自适应函数动态调整交叉和变异概率, 并在可变邻域搜索过程中利用 3 种变邻域搜索算子

提高算法的局部搜索能力。本文基于不同规模的算例进行测试, 将 VNS-NSGA-II 算法与 MOEA/D 算法、NSGA-II 算法以及 MOLNS 算法的评价指标与 Pareto 最优前沿进行了比较。研究表明, VNS-NSGA-II 算法相较于其他 3 种算法的综合性能更高, 对于小中大规模的算例, VNS-NSGA-II 算法都可以获得更好的 Pareto 最优前沿以及单个目标的最优值。与静态网络相比, TDGVRPCDTW 模型在提升客户满意度方面具有明显优势, 并且在大规模下的改进率更高, 有利于企业的长远发展。此外, 对不同碳交易价格下的总成本和碳排放进行分析, 得出适当调整碳交易价格有利于企业在减排的同时实现经济效益。

未来的研究可以考虑路面坡度的因素, 精确衡量碳排放量, 也可以改进算法提高求解效率, 还可以扩展到货运模式选择, 实现更加经济的方案。

参考文献:

- [1] LEE Y H, JUNG J W, LEE K M. Vehicle routing scheduling for cross-docking in the supply chain[J]. *Computers & Industrial Engineering*, 2006, 51(2): 247-256.
- [2] LIAO C J, LIN Y M, SHIH S C. Vehicle routing with cross-docking in the supply chain[J]. *Expert Systems with Applications*, 2010, 37(10): 6868-6873.
- [3] WEN M, LARSEN J, CLAUSEN J, et al. Vehicle routing with cross-docking[J]. *Journal of the Operational Research Society*, 2009, 60(12): 1708-1718.
- [4] SHAHABI-SHAHMIRI R, ASIAN S, TAVAKKOLI-MOGHADDAM R, et al. A routing and scheduling problem for cross-docking networks with perishable products, heterogeneous vehicles and split delivery[J]. *Computers & Industrial Engineering*, 2021, 157: 107299.
- [5] 范厚明, 白雪, 田攀俊. 集货需求可拆分的多越库中心库门分配及车辆路径协同优化 [J]. *控制与决策*, 2023, 38(2): 501-509.
- [6] CEN X K, ZHOU G, JI B, et al. Modelling and heuristically solving three-dimensional loading constrained vehicle routing problem with cross-docking[J]. *Advanced Engineering Informatics*, 2023, 57: 102029.
- [7] YAGHOUBI A, FAZLI S. Designing a new multi-objective model for the vehicle routing scheduling at a cross-docking center in mitigating CO₂ emissions at green supply chain under uncertainty[J/OL]. *Annals of Operations Research*. [2023-08-08]. <https://link.springer>.

com/article/10.1007/s10479-023-05587-w.

- [8] RAJABZADEH M, MOUSAVI S M. Allocation of products to a heterogeneous fleet of trucks in a cross-docking center based on carbon emissions and costs in food and beverage industry: novel uncertain solution approaches[J]. *Journal of Environmental Management*, 2023, 332: 117071.
- [9] ABAD H K E, VAHDANI B, SHARIFI M, et al. A bi-objective model for pickup and delivery pollution-routing problem with integration and consolidation shipments in cross-docking system[J]. *Journal of Cleaner Production*, 2018, 193: 784–801.
- [10] MALANDRAKI C, DASKIN M S. Time dependent vehicle routing problems: formulations, properties and heuristic algorithms[J]. *Transportation Science*, 1992, 26(3): 185–200.
- [11] ICHOUA S, GENDREAU M, POTVIN J Y. Vehicle dispatching with time-dependent travel times[J]. *European Journal of Operational Research*, 2003, 144(2): 379–396.
- [12] FLEISCHMANN B, GIETZ M, GNUTZMANN S. Time-varying travel times in vehicle routing[J]. *Transportation Science*, 2004, 38(2): 160–173.
- [13] HUANG Y X, ZHAO L, VAN WOENSEL T, et al. Time-dependent vehicle routing problem with path flexibility[J]. *Transportation Research Part B: Methodological*, 2017, 95: 169–195.
- [14] FIGLIOZZI M A. The time dependent vehicle routing problem with time windows: benchmark problems, an efficient solution algorithm, and solution characteristics[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2012, 48(3): 616–636.
- [15] 李楠, 胡蓉, 钱斌, 等. 时间依赖型多时间窗车辆路径问题研究 [J]. *系统仿真学报*, 2022, 34(8): 1775–1788.
- [16] LIU Y M, ROBERTO B, ZHOU J W, et al. Efficient feasibility checks and an adaptive large neighborhood search algorithm for the time-dependent green vehicle routing problem with time windows[J]. *European Journal of Operational Research*, 2023, 310(1): 133–155.
- [17] 何美玲, 杨梅, 韩珣, 等. 带时间窗的时间依赖型同时取送货车辆路径问题研究 [J]. *交通运输系统工程与信息*, 2024, 24(4): 231–242,262.
- [18] LI J K, LI J Q, XU Y. An improved multiobjective evolutionary algorithm for time-dependent vehicle routing problem with time windows[J]. *Egyptian Informatics Journal*, 2024, 28: 100574.
- [19] HICKMAN J, HASSEL D, JOUMARD R, et al. Methodology for calculating transport emissions and energy consumption[R]. Brussels: Commission of the European Communities, 1999.
- [20] MOTAGHEDI-LARIJANI A. Solving the number of cross-dock open doors optimization problem by combination of NSGA-II and multi-objective simulated annealing[J]. *Applied Soft Computing*, 2022, 128: 109448.
- [21] FANG W K, GUAN Z L, SU P Y, et al. Multi-objective material logistics planning with discrete split deliveries using a hybrid NSGA-II algorithm[J]. *Mathematics*, 2022, 10(16): 2871.

(编辑: 丁红艺)