文章编号：0253-374X(2023)08-1278-10 **DOI**：10.11908/j.issn.0253-374x.22169

考虑软时间窗的同时送取货随机旅行时间车辆路径问题

张 涛[1,2]，王楚楚[1]

(1. 上海财经大学 信息管理与工程学院，上海 200433；2. 上海市金融信息技术研究重点实验室(上海财经大学)，上海 200433)

摘要：考虑车辆总旅行时间约束和车辆载重限制以及客户对服务时间窗的要求，研究带有软时间窗的同时送取货随机旅行时间车辆路径问题(STT-VRPSPD)，建立机会约束规划模型。将禁忌搜索算法与分散搜索算法相结合，构建混合分散禁忌搜索(HSTS)算法，并采用C-W节约算法生成初始解。基于经典的Dethloff算例和Solomon时间窗生成方法，分别生成包括50个客户、200个客户各20组算例，算例测试结果验证了混合分散禁忌搜索算法的有效性。

关键词：随机旅行时间车辆路径问题(STT-VRP)；同时送取货车辆路径问题(VRPSPD)；软时间窗；混合分散禁忌搜索(HSTS)算法

中图分类号：O223 **文献标志码**：A

Stochastic Travel-time Vehicle Routing Problem with Simultaneous Pick-up and Delivery Considering Soft Time Windows

ZHANG Tao[1,2], *WANG Chuchu*[1]

(1. School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China; 2. Shanghai Key Laboratory of Financial Information Technology (Shanghai University of Finance and Economics), Shanghai 200433, China)

Abstract: According to the constraints of the total vehicle travel time, vehicle load, and the soft time windows, we studied the stochastic travel-time vehicle routing problem with simultaneous pick-up and delivery (STT-VRPSPD) considering soft time windows and established a chance-constrained programming model. Combining the tabu search algorithm with the scatter search algorithm, a hybrid scatter tabu search (HSTS) algorithm was constructed, and the C-W saving algorithm was used to generate initial solutions. Based on the classic Dethloff instances and Solomon time window generation method, 20 groups of instances including 50 customers and 20 groups of instances including 200 customers were generated respectively for numerical tests. The numerical results verify the effectiveness of the HSTS algorithm.

Key words: stochastic travel-time vehicle routing problem (STT-VRP); vehicle routing problem with simultaneous pick-up and delivery (VRPSPD); soft time windows; hybrid scatter tabu search (HSTS) algorithm

近年来，随着国内外环境保护政策的实施，制造商在安排货物运输时不仅考虑向客户正向配送货物的过程，还越来越关心从客户进行资源回收再利用的逆向物流过程。1989年，Min[1]首次提出了同时送取货车辆路径问题(VRPSPD)，随后许多学者对VRPSPD进行了扩展研究。卜雷等[2]研究了有关铁路行包配送的VRPSPD，其中每个行包配送点同时具有收货和发货业务，并设计了禁忌搜索(TS)算法对问题进行求解。Hu等[3]研究了送取货不兼容情景下的VRPSPD，并提出了一种基于可变邻域搜索的求解方法。Zachariadis等[4]研究了具有二维载荷约束的VRPSPD，并提出了一个采用记忆技术的优化框架，加速解的计算。Olgun等[5]考虑燃油消耗成本，研究了绿色VRPSPD，并开发了一种基于迭代局部搜索和可变邻域下降的超启发式算法。范厚明等[6]对VRPSPD进一步拓展，研究了多配送中心VRPSPD，其中各配送中心共享资源和车辆，协同完成送取货工作。

收稿日期：2022-04-12
基金项目：上海市科委项目(20511101403)；中央高校基本科研业务费专项资金(2023110139)
第一作者：张 涛(1970—)，男，教授，博士生导师，工学博士，主要研究方向为供应链建模与优化、生产计划与调度等。
E-mail：taozhang@mail.shufe.edu.cn
通信作者：王楚楚(1992—)，女，博士生，主要研究方向为物流优化、生产计划与调度等。
E-mail：chuchuw@163.sufe.edu.cn

论文拓展介绍

以上研究一般考虑确定的旅行速度和旅行时间，但在实际配送过程中，由于道路堵塞、天气骤变、交通事故等偶发因素常常导致车辆旅行时间无法确定，进而导致车辆无法在客户规定时间窗内进行配送回收，随机旅行时间车辆路径问题（STT-VRP）的关键即是对旅行时间的把握。Kim 等[7]提出了一个马尔可夫决策模型解决 STT-VRP，对如何估计车辆旅行时间的概率分布进行了研究。随机车辆旅行时间也可通过机会约束规划和补偿模型进行确定，可将随机车辆旅行时间转化为机会约束条件，构建带有机会约束的数学模型，建模中可将随机旅行时间构建为随机变量，并确定其分布[8]。Tas 等[9]使用伽马分布来模拟旅行时间，研究了考虑时间窗约束的 STT-VRP，提出了一种 TS 算法对问题进行求解，并指出研究中最常用的旅行时间分布为正态分布、对数正态分布、变换伽马分布和伽马分布。上述方法对随机旅行时间的潜在概率分布进行了明确设定，而当概率分布的精确信息不可得时，Zhang 等[10]在旅行时间基于经验分布的情况下，使用基于 Wasserstein 距离的模糊集描述接近经验分布的模糊分布，并研究了带时间窗的车辆路径问题。

带时间窗的车辆路径问题是符合现实物流需求的一种典型问题，能满足客户在特定时间段收发货物的倾向性，Braekers 等[11]将带时间窗车辆路径问题分为带软时间窗和带硬时间窗 2 类。Figliozzi[12]的研究表明，当硬时间窗所需的行驶路线超过可用车辆数量时，或企业需要对成本和服务进行权衡时，或有关于硬时间窗约束相对重要性的定性信息时，软时间窗约束条件更为适用。刘天虎等[13]认为，在救援中需要根据具体情况及时调整救援路线，软时间窗更符合需求，因此构建了带软时间窗的多救援车辆路径优化数学模型，并设计了结合近邻启发算法的混合遗传算法对问题进行求解。

为此，同时考虑双向物流、随机旅行时间、客户服务时间窗等因素，研究带软时间窗的同时送取货随机旅行时间车辆路径问题（STT-VRPSPD）。VRPSPD 已被证明为 NP 难题[14-15]，因此带软时间窗的 STT-VRPSPD 也是 NP 难题，许多研究采用遗传算法（GA）[16]、大邻域搜索算法[17]等启发式算法对 VRPSPD 进行求解。TS 算法也被广泛应用于车辆路径问题求解中，Lai 等[18]设计了 TS 算法求解带时间约束的车辆路径问题，有效地解决了问题中每对客户点之间具有多条并行弧带来的计算复杂性。颜瑞等[19]采用 TS 算法优化车辆路径问题，设计了 Insert、Swap、2-Opt 共 3 种局部搜索算子进行路线间和路线内的优化。Xing 等[20]提出了一种 TS 算法求解 AGV（automated guided vehicle）路径优化问题，为提高算法邻域搜索效率，对搜索过程进行了重新定位和交换操作。分散搜索（SS）将决策规则和约束条件结合来寻求问题的解决方案，是一种基于局部搜索的启发式算法，适用于复杂优化问题的求解。Alinaghian 等[21]采用结合变邻域搜索的 SS 算法解决了动态灾后空中救援车辆路径问题。Soman 等[22]考虑异构车队车辆路径问题，设计了一种具有策略性振荡的 SS 算法来解决该问题。SS 的特点是利用参考集保证整个搜索算法的广泛性。由于子集组合产生新解的过程较复杂且耗时，为了提高算法性能，经常将 SS 的进化策略与 TS 的自适应记忆技术相结合[23]。Russell 等[24]设计了一种混合分散禁忌搜索（HSTS）算法求解带硬时间窗的车辆路径问题，算法通过生成邻域结构并禁忌路径中不满足约束的节点移动，保证搜索过程的有效性和多样性。Ge 等[25]将 TS 动态禁忌机制嵌入 SS 框架中，求解带软时间窗的车辆路径问题。

为了解决带软时间窗的 STT-VRPSPD，将软时间窗约束引入机会约束规划中，对随机旅行时间进行静态化处理，考虑软时间窗约束对车辆旅行总时间的影响，并对数学模型进行修正。构造 TS 与 SS 相结合的混合分散禁忌搜索算法，采用 C-W 节约值算法改进 SS 算法的初始解，通过 TS 算法对 SS 算法进行局部优化，提升 SS 算法的全局寻优性能。

1　带软时间窗的 STT-VRPSPD 模型

带软时间窗的 STT-VRPSPD 描述如下：完全网络 $G=(V^*,A)$，顶点集合 $V^*=\{v_0,v_1,\cdots,v_n\}$，顶点 v_0 表示中心仓库，$V=V^*/\{v_0\}$ 表示客户集合，V 中各客户同时具有送货需求和取货需求，且均具有服务时间窗 $[E_j,L_j]$；车辆在弧 (v_i,v_j) 上的旅行时间不是固定的，而是分布已知的随机变量；假设车辆的载重有限，客户服务时间要求为软时间窗，每辆车均具有最大旅行时间的限制；不考虑车辆在客户点完成装卸服务所需要的时间；车辆从中心仓库出发服务客户的送取货需求，当其无法满足负载或旅行时间的约束时则返回中心仓库，由其他车辆继续服务剩余客户；每个客户只能被一辆车服务且只能被服务一次。在满足所有客户送取货需求、软时间窗约束和车辆负载及最大总旅行时间限制的条件下，确

定期望总成本最小化的车辆路径。假设如下:①一个中心仓库;②中心仓库与客户的位置坐标已知;③客户的送取货需求已知;④客户服务时间窗已知;⑤每段路径上的旅行时间服从正态分布,具体分布已知;⑥车辆负载不超过最大限制;⑦需满足所有客户的需求;⑧客户同时送取货物;⑨所有车辆车型一致且负载约束已知;⑩每个客户必须且只能被服务一次。为了表述模型,设置符号体系,如表1所示。

表1 符号定义

Tab.1 Definition of symbols

变量	定义
V	所有客户的集合,$V=\{v_1,v_2,\cdots,v_n\}$
V^*	所有顶点的集合,$V^*=\{v_0\}\cup V$,其中v_0代表中心仓库
m	可供使用的最大车辆数
n	客户的个数
(v_i,v_j)	车辆从顶点i到顶点j行驶的路径
$[E_j,L_j]$	时间窗,E_j为客户j的时间窗开启时间,L_j为客户j的时间窗关闭时间
$\boldsymbol{\xi}$	一组随机向量,其中每个分量对应车辆在(v_i,v_j)路径上的旅行时间
Ξ	$\boldsymbol{\xi}$的样本空间,是有限样本空间
$t_{ijl\boldsymbol{\xi}}$	状态$\boldsymbol{\xi}$下,车辆l在路径(v_i,v_j)上的旅行时间
v_{ijl}	车辆l在路径(v_i,v_j)上的平均行驶速度
D_{ij}	顶点i到顶点j的距离,其值等于速度和旅行时间的乘积,$D_{ij}=v_{ijl}t_{ijl\boldsymbol{\xi}}$
c_{ijl}	车辆l在路径(v_i,v_j)上的运行成本,在本实验中,c_{ijl}取与D_{ij}相同的值
B_l	车辆l每日总旅行时间的上界
d_i	客户i的送货需求量
p_i	客户i的取货需求量
Q_l	车辆l的最大负载能力
q_{ijl}	车辆l访问完客户i后,在访问客户j之前的负载量
α	车辆l的总旅行时间不超过给定上界B_l的概率
β	提前到达的惩罚系数
γ	滞后到达的惩罚系数
t_{jl}	车辆l抵达第j个客户点的时刻
G	单位发车成本
$x_{ijl}=\begin{cases}1\\0\end{cases}$	车辆l从客户i到客户j,该值为1;其他,为0
$z_{il}=\begin{cases}1\\0\end{cases}$	客户i由车辆l服务,该值为1;其他,为0

带软时间窗的车辆路径问题中,若车辆l在E_j前到达客户j,则车辆需在此等待,产生机会损失成本;若车辆l在L_j后到达客户j,则服务被延误,产生惩罚成本。惩罚函数如下所示:

$$f(t_{jl})=\begin{cases}\beta z_{jl}(E_j-t_{jl}),t_{jl}<E_j\\0,E_j\leqslant t_{jl}\leqslant L_j\\\gamma z_{jl}(t_{jl}-L_j),t_{jl}>L_j\end{cases}\tag{1}$$

带软时间窗STT-VRPSPD的旅行时间是随机的,车辆l的总旅行时间也是随机的,并且不超过给定的上界B_l,这实际上可视为一个机会约束条件。在随机旅行时间的观测值实现之前,求解期望成本最低的车辆行驶路径,构建带软时间窗STT-VRPSPD的机会约束规划模型,如下所示:

$$\min\sum_{l=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n}c_{ijl}x_{ijl}+\beta\sum_{l=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n}x_{ijl}\max\{E_j-t_{jl},0\}+\gamma\sum_{l=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n}x_{ijl}\max\{t_{jl}-L_j,0\}+\sum_{l=1}^{m}\sum_{i=1}^{n}x_{0il}G\tag{2}$$

s.t.

$$\sum_{l=1}^{m}\sum_{i=0}^{n}x_{ijl}=1,j=1,2,\cdots,n\tag{3}$$

$$\sum_{i=0}^{n}x_{ijl}-\sum_{i=0}^{n}x_{jil}=0,j=1,2,\cdots,n,l=1,2,\cdots,m\tag{4}$$

$$\sum_{j=1}^{n}x_{0jl}+\sum_{j=1}^{n}x_{j0l}=2z_{0l},l=1,2,\cdots,m\tag{5}$$

$$\sum_{i=1}^{n}x_{0il}\leqslant 1,l=1,2,\cdots,m\tag{6}$$

$$P\left(\sum_{i=0}^{n}\sum_{j=0}^{n}t_{ijl\boldsymbol{\xi}}x_{ijl}+\sum_{j=1}^{n}z_{jl}\max\{E_j-t_{jl},0\}\leqslant B_l\right)\geqslant\alpha,l=1,2,\cdots,m\tag{7}$$

$$\sum_{i\in S}\sum_{j\in S}x_{ijl}\leqslant|S|-1,\ S\subseteq V,l=1,2,\cdots,m\tag{8}$$

$$\sum_{j=1}^{n}q_{0jl}=\sum_{i=0}^{n}\sum_{j=1}^{n}x_{ijl}d_j,l=1,2,\cdots,m\tag{9}$$

$$\sum_{l=1}^{m}\sum_{i=0}^{n}q_{ijl}x_{ijl}-d_j=\sum_{l=1}^{m}\sum_{i=0}^{n}q_{jil}x_{jil}-p_j,j=0,1,2,\cdots,n \tag{10}$$

$$0\leqslant q_{ijl}\leqslant Q_l,i=0,1,2,\cdots,n,j=0,1,2,\cdots,n,l=1,2,\cdots,m \tag{11}$$

$$z_{il}\in\{0,1\},i=0,1,2,\cdots,n,l=1,2,\cdots,m \tag{12}$$

$$x_{ijl}\in\{0,1\},i=0,1,2,\cdots,n,j=0,1,2,\cdots,n,l=1,2,\cdots,m \tag{13}$$

式(7)、(8)中:P为事件成立的概率;S为集合V的真子集;$|S|$为集合S中客户点的个数。式(2)是目标函数,即最小化总车辆行驶成本、提前/延误服务惩罚和发车成本;式(3)表明每个客户均被服务一次且仅一次;式(4)确保车辆l服务客户j后也从该客户点离开;式(5)表示每辆车从中心仓库出发并最终返回;式(6)表示每辆车最多只能使用一次;式(7)保证车辆l的总工作时间小于B_l的概率大于α;式(8)排除奇异子回路;式(9)表示车辆l出发时的载货量为所服务客户的送货需求量之和;式(10)表示车辆服务客户j的前后载货量变化;式(11)表示每辆车的载货量范围;式(12)、(13)表示变量z_{il}、x_{ijl}的取值范围。

带软时间窗STT-VRPSPD的机会约束模型可以根据给定的置信水平,将机会约束转化为确定约束,随后求解带确定软时间窗约束的STT-VRPSPD。假设车辆l在每条弧上的旅行时间相互独立,并且在弧(v_i,v_j)上的旅行时间服从正态分布$N(D_{ij}/v_{ijl},\sigma_{ijl}^2)$,其中,$\sigma_{ijl}$为车辆$l$在弧$(v_i,v_j)$上旅行时间的标准差,故$B_l-\sum_{i=0}^{n}\sum_{j=0}^{n}t_{ijl\xi}x_{ijl}$也是随机变量,并服从正态分布$N\left(B_l-\sum_{i=0}^{n}\sum_{j=0}^{n}D_{ij}x_{ijl}/v_{ijl},\sum_{i=0}^{n}\sum_{j=0}^{n}\sigma_{ijl}^2x_{ijl}^2\right)$,则机会约束(7)等价于式(14)。

$$P\left\{\eta\leqslant\frac{B_l-\sum_{i=0}^{n}\sum_{j=0}^{n}D_{ij}x_{ijl}/v_{ijl}-\sum_{j=1}^{n}z_{jl}\max\{E_j-t_{jl},0\}}{\sqrt{\sum_{i=0}^{n}\sum_{j=0}^{n}\sigma_{ijl}^2x_{ijl}^2}}\right\}\geqslant\alpha \tag{14}$$

式中:η为随机变量,服从标准正态分布。式(14)成立当且仅当

$$\Phi^{-1}(\alpha)\leqslant\frac{B_l-\sum_{i=0}^{n}\sum_{j=0}^{n}D_{ij}x_{ijl}/v_{ijl}-\sum_{j=1}^{n}z_{jl}\max\{E_j-t_{jl},0\}}{\sqrt{\sum_{i=0}^{n}\sum_{j=0}^{n}\sigma_{ijl}^2x_{ijl}^2}} \tag{15}$$

式中:Φ为标准正态分布概率密度函数。t_{jl}满足下式:

$$t_{jl}x_{ijl}=(t_{il}+\max\{E_i-t_{il},0\}+t_{ijl\xi})x_{ijl} \tag{16}$$

2　HSTS算法设计

2.1　编码

采用自然数编码,用1~n间的自然数表示n个客户点,这些自然数的任意排列形成长度不变的数字串。在代表客户点全排列的数字串中插入代表中心仓库的数字零,表示各条车辆路径的分界点,这样每个数字串都代表一个解。

2.2　初始解集生成方法

SS算法对初始解的依赖性较强,所以算法首先需要产生质量较高的多样性初始解。SS算法中通常采用Glover提出的多样性产生器生成初始解,但这只能在一定程度上增加解的多样性。

本研究采用节约算法生成初始解,步骤具体为:首先将每个客户点与中心仓库一一连接,共产生N条路径,然后利用三角不等式的概念,将每两客户点合并至同一路线所产生的距离节约值按降序排列,并依次生成线路,逐步连接,直至全部节约值均不大于零且不违背容量约束为止,得到问题的最终解。

由于软时间窗与硬时间窗的约束不同,是一个比较宽松的约束机制,仅在惩罚成本上对路径选择进行约束,因此节约值计算式可以不考虑时间窗因素。

基于上述考虑,定义如下节约值计算式:

$$S'(i,j)=t_{0il\xi}+t_{j0l\xi}-t_{ijl\xi} \tag{17}$$

$$S^*(i,j)=\begin{cases}S'(i,j)+\omega(d_j-p_j),d_j>p_j\\S'(i,j)-\varphi(d_j-p_j),d_j\leqslant p_j\end{cases} \tag{18}$$

式中:$S'(i,j)$为加入节点j所能增加的时间节约值;ω为节约值正相关系数,$\omega(d_j-p_j)$表示路线中加入节点j所增加的车辆装载效益,将这类节点放在路线的前部是有益的,与节约值正相关;φ为节约值负相关系数,$\varphi(d_j-p_j)$表示路线中加入节点j所损失的车

辆装载效益,这类节点放在路线的后部对路线才有益,因此与节约值负相关。

2.3 参考集更新方法

参考集大小为$|b|$,由2类解构成,一类是高质量解集$|b_1|$,另一类是多样性解集$|b_2|$,在这里取$|b|=|b_1|+|b_2|$。高质量解为目标值较优的解,多样性解根据其与高质量解集的距离选取,解间距离表示它们之间的差异大小。

根据目标值的大小,按从小到大的顺序对多样性产生方法或解改进方法产生的解排序,并取其中前$|b_1|$个解组成高质量解参考集。计算其余各解与高质量解集的距离,取距离最大的前$|b_2|$个解组成多样性解参考集。2个解间的距离为max{各解中的弧数}−2个解中共有的弧数,解与高质量解参考集的距离为min{解与各高质量解的距离}。

2.4 子集产生方法

通过子集产生方法,可以为下一步解的组合产生更多更好的解,但受制于运行速度的限制,设定子集总数需等同于初始种群中解的数量,每个子集通过组合参考集中任意不同的2个解产生。具体步骤如下:

(1)参考集中的解两两组合形成$C_{|b|}^2$子集。

(2)随机选择K个子集组合成为“解的组合方法”操作的子集群。

2.5 解的组合方法

组合子集中的解产生新解,新解既可以保留上一代解的优良特性,又可以增加解的多样性,跳出局优。解的组合方式为:在子集中的2个原解上随机选择同一位置作为交换点,保留其中一个解交换点之前的片段,并删去另一解中已保留的客户点,将另一解中其余的客户点按原顺序排列于保留片段之后。例如,解1:987|456321和解2:123|467895(“|”表示交换点的位置),那么保留解1前段可以得到新解1:987123465,保留解2前段可以得到新解2:123987456。比较原解1、原解2、新解1、新解2的目标值大小,保留目标值最小的那个解。

2.6 TS改进策略

为了避免SS算法过早陷入局优,提升算法的爬山能力,将TS算法与SS算法相融合,TS算法采用2-opt变换进行局部改进。

2-opt变换的操作为:在线路内任取2个不相同的节点,将这2个节点之间所有客户点的车辆行驶方向进行逆转,如原路径中选中2、5 2个点,具体变换过程如图1所示。

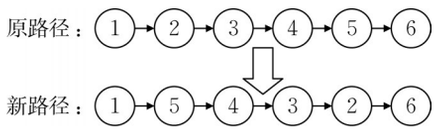


图1 2-opt变换示意图

Fig.1 Schematic diagram of 2-opt exchange

TS算法改进流程如下:

(1)将当前解集中所有解作为禁忌搜索的初始解,对解集中所有解按P_t概率判断是否进行禁忌搜索解改进。

(2)置空禁忌表,禁忌迭代次数置零。

(3)根据2-opt变换选出最佳解作为候选解,禁忌搜索迭代次数$R=R+1$,判断候选解是否已被禁忌。若候选解被禁忌且不符合特赦条件,则重新执行(3),否则转至(4)。

(4)比较候选解与当前最优解目标值的大小,若候选解小于最优解,则更新最优解。

(5)判断是否达到禁忌搜索最大迭代次数,若已达到则执行(6),否则跳转至(3)。

(6)输出禁忌搜索改进后的解,更新当前路径。

2.7 HSTS算法流程

提出的HSTS算法流程如图2所示。

3 实验分析

3.1 实验数据生成

应用Dethloff随机测试算例的生成方法随机生成包含50客户点、200客户点的样本分组。Dethloff随机测试算例的生成过程按顾客的地理布局,可分为2种不同的布局模式。模式一SCA类,顾客的坐标一律分布在区间[0,100];模式二CON类,一半顾客与SCA类的分布方法相同,另一半顾客的坐标更加集中地分布在区间[100/3,200/3]。在200客户点样本生成时将分布区间扩大到[0,200],相应地,CON类集中区间为[200/3,400/3]。地理布局的距离测量采用欧几里得矩阵。顾客需求的送货量d_i是分布在区间[0,100]的整数;顾客需求的取货量p_i(整数)通过公式$p_i=(0.5+r_i)d_i$计算而得,其中r_i是[0,1]上的随机数。车辆负载限制$Q=\sum_{i=1}^{n}d_i/\mu$,其中μ代表所选的最小车辆数。选取μ为3,SCA3和CON3作为算例组。

假设车辆在每条弧上的旅行时间$t_{ij\xi}$相互独立,

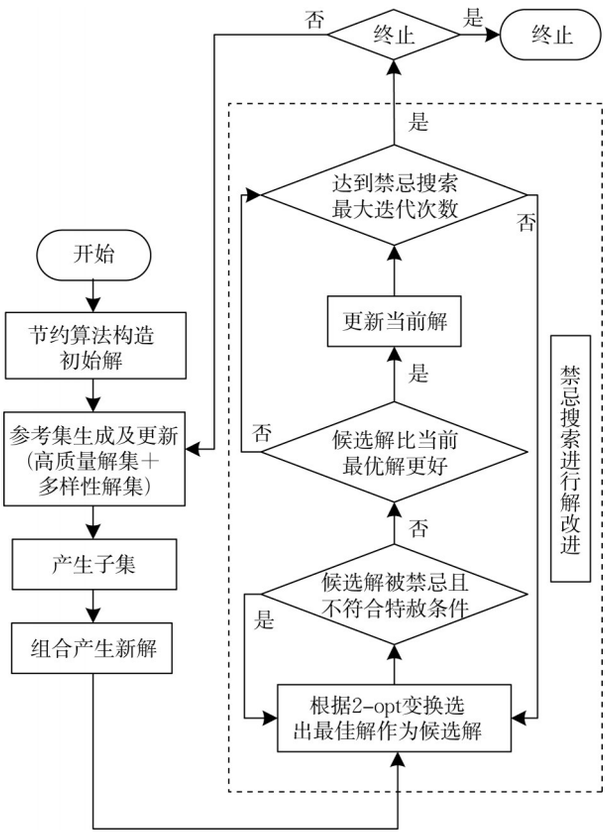


图2　HSTS算法流程

Fig.2　Flowchart of the HSTS algorithm

并且在弧(v_i,v_j)上的旅行时间服从均值为D_{ij}/v_{ijl}、标准差σ为1的正态分布;车辆的平均行驶速度v_{ijl}为40;每辆车的旅行时间上界为B;D_{ij}为顶点v_i和v_j间的距离,作为车辆在弧(v_i,v_j)上的旅行成本。

实验数据的时间窗参照Solomon[26]提出的带时间窗车辆路径(VRPTW)基准数据生成方法,并稍做调整,生成规则为:首先,假设所有客户点均具有时间窗,客户点不存在服务时间;然后,通过随机值排序选定获得时间窗的各个客户点;最后,在区间$(E_0+t_{0il\xi},L_0-t_{i0l\xi})$生成客户点的时间窗中间值,时间窗宽度半径为正态分布随机数。其中,E_0为中心仓库开始服务的时间,L_0为中心仓库结束服务的时间,$t_{0il\xi}$和$t_{i0l\xi}$分别表示车辆l从中心仓库到客户i的旅行时间和从客户i回到中心仓库的旅行时间。

3.2　模型参数设置

实验中,软时间窗惩罚系数β和γ均取为1;发车成本G取额定值为50。

GA算法为最常用于车辆路径问题及其变体问题研究的元启发式算法[27]。选取GA算法对STT-VRPSPD模型进行检验,测试单一车辆旅行总时间小于车辆旅行时间上界的概率α和车辆旅行时间上界B的组合。GA算法交叉概率为0.6,变异概率为0.1,群体规模为100,运行代数为15 000。

概率α取值分别为0.7、0.8、0.9,车辆旅行时间上界B分别取值为23、24、25、26,分别在α和B共3×4=12组参数组合下,对SCA3-0算例测试10次并统计平均值,结果如图3所示。

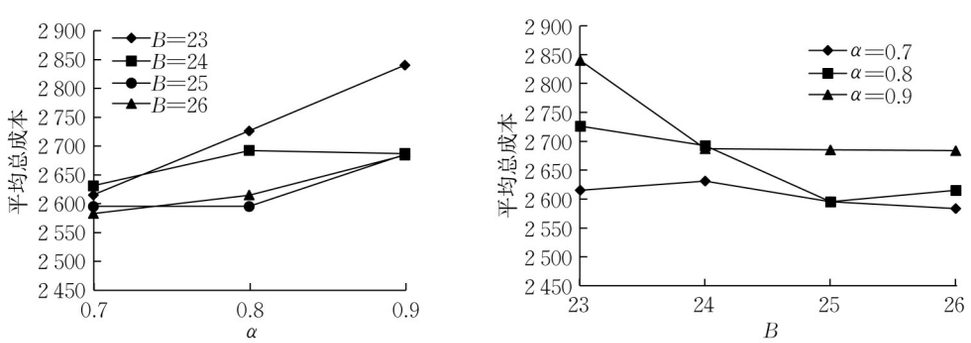


图3　平均总成本随B、α变动

Fig.3　Variation of average total cost with B and α

由图3发现,B相同时,平均总成本随α的增大呈现增加的趋势。这是由于随着α的增加,车辆旅行时间上界的约束变得更加严格,需要更多的车辆才能完成配送任务,因此总成本升高。α相同时,实验结果大致符合平均总成本随B值增大而减小的趋势。这是因为B增大,车辆的最大旅行时间约束变得更加宽松,单辆车能够服务更多的客户点,使得总发车数量减少,与之相对应的发车成本也减少。

B达到25后,随着B继续增大,平均总成本变动较为平缓;同样,α=0.8后,随着α的继续减小,平均总成本变动较为平缓。为了防止约束条件过分宽松而导致车辆旅行时间上限的约束失效,选择参数组合α=0.8、B=25进行后续实验。

3.3　HSTS算法重要参数分析

在模型参数α=0.8、B=25的情况下,50客户点样本数据的SCA3-0算例中对HSTS算法的不同参

数组合分别进行10次实验,比较实验的平均结果, 如表2所示。

表2 不同参数组合下的实验结果(SCA3-0算例)

Tab.2 Experimental results under different combinations of parameters(dataset SCA3-0)

编号	K	b	b_1	b_2	总成本	车辆数	达到最好解所需平均计算时间/s	达到最好解所需平均迭代次数
1	100	15	10	5	1 868	7	37.60	16
2	100	15	5	10	1 802	7	35.02	14
3	100	10	5	5	1 930	8	12.45	5
4	100	20	10	10	1 722	6	43.50	16
5	100	20	15	5	1 865	7	27.80	12
6	100	20	5	15	1 800	7	35.42	15
7	50	20	10	10	1 806	7	22.75	23
8	150	20	10	10	1 669	7	62.46	16

参数主要选取K、b、b_1、b_2的多种组合进行测试,由于$b=b_1+b_2$,也可以认为是3个参数K、b_1、b_2的多组合测试。其中1~6组主要针对参数b_1、b_2进行测试,由4~6组测试发现,在b值确定时,$b_1=b_2$情况下总成本和车辆数都达到最小。

根据前6组测试结果,在K值确定时,总成本和车辆数随b值的增加而呈现下降趋势,这是因为b值增加时,更多的多样性解或优质解被纳入参考集中,产生的新解的多样性和优良性都更有保证。同时,对比4、7、8组测试结果发现,当参数b、b_1、b_2值均确定时,K值的增加会使总成本和车辆数呈现下降趋势。这是因为随着K值的增加,解空间被扩大,单次搜索的范围也更加广阔,所得解的质量也能得到提高。因此,参数组K=150、b=20、b_1=10、b_2=10时,算法效果在所有参数组合中最好。

然而,各参数组合条件下算法的运行时间也是本研究选择参数值的重要考量因素。根据实验结果可知,随着K值的增加平均总成本降低,但是计算时间随之增加。计算发现,当K=150时测试所需时间比K=100时增加了43.6%,而总成本仅下降3.1%。综合考虑各因素,选择K=100、b=20、b_1=10、b_2=10作为HSTS算法的实验参数值。

3.4 HSTS算法与GA算法实验结果对比分析

(1)50客户点样本

分别采用GA算法和HSTS算法求解50客户点的CON3和SCA3 2类共20组算例,每组测试10次并统计平均值,如表3所示。

由表3看出,提出的HSTS算法能够求得更优质的可行解,平均总成本比GA算法所得结果改进了30.83%,HSTS算法的最好值、最差值也都较GA算法对应值更小。计算速度方面,GA算法所需计算时间约为HSTS算法的2倍,HSTS算法计算效率更高。

(2)200客户点样本

分别采用GA算法和HSTS算法求解200客户点的CON3和SCA3 2类共20组算例,每组测试10次并统计平均值,如表4所示。

由表4看出,在运行于大规模样本时GA算法收敛慢、易陷入局优的缺陷更加明显。HSTS算法能够更快速地得到更优质的可行解,平均总成本比GA算法所得结果改进了41.22%,平均计算时间比GA算法缩减了15.51%。200客户点算例优化结果中,需要调用的车辆数一般在25至35辆范围内。GA算法平均调用车辆为32辆,而HSTS算法平均调用车辆为27辆,大大减少了总发车数量。

依据GA算法、HSTS算法分别求解50客户点算例中最优的一次结果,绘制折线图4、5观察2种算法对车辆负载约束和车辆旅行时间约束的满足情况。GA算法求得的结果用到了5辆车,HSTS算法用到了4辆车,都能实现所求问题的可行解。

由图4可见,HSTS算法结果的前3辆车辆负载量都得到了较充分的应用,GA算法得到的结果则在车辆负载量使用上不及HSTS算法充分,仅第4辆车能够达到较高的利用率。

图5为GA算法和HSTS算法车辆旅行时间。可以发现,GA算法的第2~4辆车均能逼近期望旅行时间上限,除最后一辆车不饱和外,其他车辆的旅行时间都达到20以上,此时旅行时间约束成为该车辆行驶的主要约束。HSTS算法的结果较为均匀,且各车辆的总旅行时间均在20左右,距离各车期望旅行时间上界有一定的差距,但仍处于可接受范围内。这里,期望旅行时间上界为$B_l-\Phi^{-1}(\alpha)\sqrt{\sum_{i=0}^{n}\sum_{j=0}^{n}\sigma_{ijl}^2x_{ijl}^2}$。

由于引入了软时间窗约束,并将其纳入问题模型,因此在讨论实验结果时不应只考虑期望旅行时

表 3　GA 算法和 HSTS 算法求解 SCA3 和 CON3 算例结果的对比(50 客户点)

Tab.3　Comparison of results between GA algorithm and HSTS algorithm for solving dataset SCA3 and CON3 (50 customer points)

编号	GA 算法					HSTS 算法					改进率/%
	最好值	最差值	均值	计算时间/s	车辆数	最好值	最差值	均值	计算时间/s	车辆数	
SCA3-0	2 570	2 630	2 595	73.33	6	1 698	1 743	1 722	43.50	6	33.64
SCA3-1	2 520	2 631	2 582	79.25	7	1 688	1 754	1 722	37.84	6	33.31
SCA3-2	2 637	2 750	2 679	83.09	7	1 467	1 630	1 574	51.81	6	41.25
SCA3-3	1 963	2 484	2 506	48.95	6	1 766	1 887	1 820	40.96	6	27.37
SCA3-4	2 716	2 777	2 751	78.59	7	1 712	1 826	1 761	44.06	6	35.99
SCA3-5	2 412	2 574	2 466	86.36	5	1 776	1 792	1 786	42.32	7	27.58
SCA3-6	2 366	2 478	2 421	62.86	5	1 494	1 704	1 584	52.38	6	34.57
SCA3-7	2 418	2 590	2 524	49.42	5	1 562	1 640	1 596	69.16	6	36.77
SCA3-8	2 221	2 398	2 315	70.04	6	1 467	1 555	1 504	50.45	6	35.03
SCA3-9	2 557	2 588	2 569	148.28	6	1 595	1 805	1 725	49.29	6	32.85
CON3-0	1 854	1 973	1 883	93.17	6	1 382	1 476	1 421	48.18	6	24.54
CON3-1	1 873	1 982	1 930	81.22	6	1 347	1 450	1 408	56.68	6	27.05
CON3-2	2 011	2 098	2 005	75.05	6	1 420	1 503	1 448	47.36	6	27.78
CON3-3	1 970	2 038	2 011	98.28	6	1 370	1 488	1 437	49.88	6	28.54
CON3-4	2 024	2 115	2 073	81.79	6	1 455	1 592	1 538	46.31	7	25.81
CON3-5	1 957	1 986	1 973	99.72	5	1 308	1 421	1 362	45.32	5	30.97
CON3-6	1 953	2 020	1 991	90.85	5	1 407	1 538	1 494	53.88	6	24.96
CON3-7	1 949	1 999	1 981	139.70	5	1 453	1 504	1 482	43.43	7	25.19
CON3-8	1 632	1 705	1 662	67.08	5	1 120	1 221	1 182	41.41	4	28.88
CON3-9	1 981	2 031	2 007	93.99	5	1 272	1 386	1 315	48.16	4	34.48
平均值	2 179	2 292	2 246	85.05		1 488	1 596	1 544	48.12		30.83

表 4　GA 算法和 HSTS 算法求解 SCA3 和 CON3 算例结果的对比(200 客户点)

Tab.4　Comparison of results between GA algorithm and HSTS algorithm for solving dataset SCA3 and CON3 (200 customer points)

编号	GA 算法					HSTS 算法					改进率/%
	最好值	最差值	均值	计算时间/s	车辆数	最好值	最差值	均值	计算时间/s	车辆数	
SCA3-0	16 832	17 037	16 915	772.76	34	9 645	9 851	9 738	707.11	23	42.43
SCA3-1	16 489	17 025	16 769	771.23	32	11 409	12 994	12 266	293.69	30	26.85
SCA3-2	16 286	18 213	16 955	756.85	33	9 825	10 093	9 981	760.16	26	41.13
SCA3-3	16 213	17 732	16 983	799.48	34	11 693	12 373	11 938	351.36	30	29.71
SCA3-4	15 640	18 642	16 835	734.15	33	9 660	10 237	9 856	600.67	28	41.46
SCA3-5	15 892	17 399	16 708	803.36	33	10 341	10 511	10 430	719.49	29	37.57
SCA3-6	16 773	18 327	17 494	791.26	34	9 817	10 352	10 021	731.11	29	42.72
SCA3-7	16 362	18 303	17 300	778.37	33	9 957	11 294	10 542	711.99	28	39.06
SCA3-8	16 824	17 231	17 091	798.07	33	9 877	10 735	10 437	744.67	28	38.93
SCA3-9	16 235	17 021	16 696	795.28	32	9 748	10 734	10 152	709.19	27	39.20
CON3-0	14 797	16 388	15 506	828.72	31	9 241	9 490	9 404	742.99	28	39.35
CON3-1	15 935	16 965	16 412	819.34	33	8 970	9 076	9 015	730.56	26	45.07
CON3-2	15 376	16 623	16 108	765.71	33	8 345	8 758	8 553	625.55	24	46.90
CON3-3	15 892	16 632	16 251	776.71	32	8 838	9 777	9 244	652.22	26	43.12
CON3-4	16 047	16 742	16 477	781.89	32	8 783	8 984	8 902	704.99	27	45.97
CON3-5	14 723	16 032	15 517	790.89	33	9 158	9 329	9 232	751.84	26	40.50
CON3-6	15 964	16 529	16 321	809.00	33	8 837	9 462	9 011	705.97	26	44.79
CON3-7	16 264	16 935	16 513	754.16	32	8 725	8 866	8 812	672.87	26	46.64
CON3-8	16 031	17 395	16 734	800.80	31	8 683	9 648	9 188	711.44	27	45.09
CON3-9	16 383	17 003	16 638	822.53	33	8 257	8 921	8 670	680.30	26	47.89
平均值	16 048	17 209	16 611	787.53		9 490	10 074	9 770	665.41		41.22

间上界的满足情况,还应考虑由于不满足软时间窗约束而产生的惩罚。GA 算法中,由于不满足软时间窗约束而产生的惩罚值总计为 246.03。HSTS 算法中,由于不满足软时间窗约束而产生的惩罚值总计为 178.32。GA 算法中第 2~4 辆车虽然基本达到期望旅行时间上界,但都需要承担较高的未按时服务惩罚值,而 HSTS 算法仅第 1 辆车的未按时服务惩罚值较高。

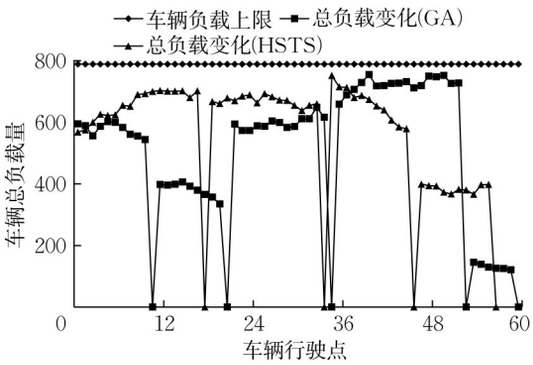
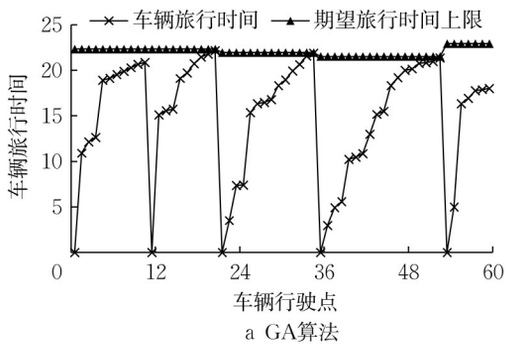
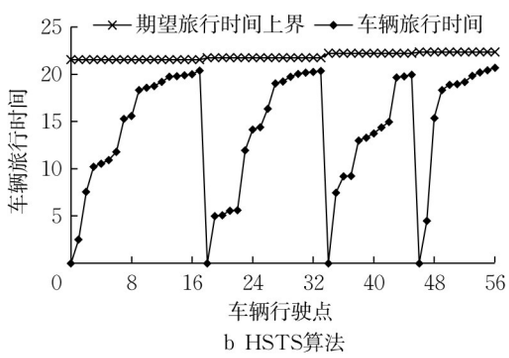


图4 车辆负载情况对比

Fig.4 Comparison of vehicle load conditions



a GA算法



b HSTS算法

图5 GA算法和HSTS算法车辆旅行时间

Fig.5 Vehicle travel time under GA algorithm, HSTS algorithm

4 结语

研究了带软时间窗的STT-VRPSPD,并建立了带软时间窗的STT-VRPSPD模型。将禁忌搜索和分散搜索算法相结合,构造了HSTS算法。根据Dethloff算例和Solomon算例生成规则并加以扩充,生成了多组算例,将HSTS算法与GA算法进行对比实验,验证了HSTS算法可以更有效地求解带软时间窗的STT-VRPSPD。随着测试算例规模的增加,HSTS算法优越性更加明显。未考虑车辆在客户点完成服务所需要的时间,而该时间在实际问题中对装卸时间的控制很重要,后期可以进一步深入研究。

作者贡献声明:

张 涛:问题提出,论文写作和修改指导。

王楚楚:问题建模,算法设计,实验仿真,论文写作。

参考文献:

[1] MIN H. The multiple vehicle routing problems with simultaneous delivery and pick-up points [J]. Transportation Research A, 1989, 23(5): 377.

[2] 卜雷, 尹传忠, 赵宜. 铁路行包配送车辆路径问题模型及算法[J].同济大学学报(自然科学版), 2007(8): 1069.
BU Lei, YIN Chuanzhong, ZHAO Yi. Model and algorithm of vehicle routing problem for railway parcel distribution [J]. Journal of Tongji University(Natural Science), 2007(8): 1069.

[3] HU Z H, SHEU J B, ZHAO L, *et al*. A dynamic closed-loop vehicle routing problem with uncertainty and incompatible goods [J]. Transportation Research, Part C: Emerging Technologies, 2015, 55: 273.

[4] ZACHARIADIS E E, TARANTILIS C D, KIRANOUDIS C T. The vehicle routing problem with simultaneous pick-ups and deliveries and two-dimensional loading constraints [J]. European Journal of Operational Research, 2016, 251(2): 369.

[5] OLGUN B, KOC C, ALTIPARMAK F. A hyper heuristic for the green vehicle routing problem with simultaneous pickup and delivery [J]. Computers & Industrial Engineering, 2021, 153: 107010.

[6] 范厚明, 张轩, 任晓雪, 等. 多中心开放且需求可拆分的VRPSDP问题优化[J]. 系统工程理论与实践, 2021, 41(6): 1521.
FAN Houming, ZHANG Xuan, REN Xiaoxue, *et al*. Optimization of multi-depot open split delivery vehicle routing problem with simultaneous delivery and pick-up [J]. Systems Engineering: Theory & Practice, 2021, 41(6): 1521.

[7] KIM G, ONG Y S, CHEONG T, *et al*. Solving the dynamic vehicle routing problem under traffic congestion [J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(8): 1.

[8] PILLAC V, GENDREAU M, GUÉRET C, *et al*. A review of dynamic vehicle routing problems [J]. European Journal of Operational Research, 2013, 225(1): 1.

[9] TAS D, DELLAERT N, WOENSEL T V, *et al*. Vehicle routing problem with stochastic travel times including soft time windows and service costs [J]. Computers & Operations Research, 2013, 40: 214.

[10] ZHANG Y, ZHANG Z Z, LIM A, *et al*. Robust data-driven vehicle routing with time windows [J]. Operations Research, 2021, 69(2): 469.

[11] BRAEKERS K, RAMAEKERS K, NIEUWENHUYSE I V. The vehicle routing problem: state of the art classification and

review [J]. Computers & Industrial Engineering, 2015, 99: 300.

[12] FIGLIOZZI M A. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows [J]. Transportation Research, Part C: Emerging Technologies, 2010, 18(5): 668.

[13] 刘天虎, 许维胜, 吴启迪. 突发灾害下带软时间窗多车路径搜索建模[J]. 同济大学学报(自然科学版), 2012, 40(1):109.
LIU Tianhu, XU Weisheng, WU Qidi. Multi-vehicle routing search modeling with soft time windows under sudden disasters [J]. Journal of Tongji University(Natural Science), 2012, 40(1):109.

[14] DETHLOFF J. Relation between vehicle routing problems: an insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls [J]. Journal of the Operational Research Society, 2002, 53(1): 115.

[15] ANILY S. The vehicle-routing problem with delivery and back-haul options [J]. Naval Research Logistics, 1996, 43(3): 415.

[16] ZHANG W Y, CHEN Z X, ZHANG S, *et al*. Dynamic multi-stage failure-specific cooperative recourse strategy for logistics with simultaneous pickup and delivery [J]. Soft Computing, 2020, 25(5): 3795.

[17] HOF J, SCHNEIDER M. An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery [J]. Networks, 2019, 74(3): 207.

[18] LAI D S W, DEMIRAG O C, LEUNG J M Y. A tabu search heuristic for the heterogeneous vehicle routing problem on a multigraph [J]. Transportation Research, Part E: Logistics and Transportation Review, 2016, 86: 32.

[19] 颜瑞, 陈立双, 朱晓宁, 等. 考虑区域限制的卡车搭载无人机车辆路径问题研究[J]. 中国管理科学, 2022, 30(5): 144.
YAN Rui, CHEN Lishuang, ZHU Xiaoning, *et al*. Study on vehicle routing problem of truck-mounted UAV considering area restrictions [J]. Chinese Journal of Management Science, 2022, 30(5): 144.

[20] XING L N, LIU Y Y, LI H Y, *et al*. A novel tabu search algorithm for multi-AGV routing problem [J]. Mathematics, 2020, 8(2): 279.

[21] ALINAGHIAN M, AGHAIE M, SABBAGH M S. A mathematical model for location of temporary relief centers and dynamic routing of aerial rescue vehicles [J]. Computers & Industrial Engineering, 2019, 131: 227.

[22] SOMAN J T, PATIL R J. A scatter search method for heterogeneous fleet vehicle routing problem with release dates under lateness dependent tardiness costs [J]. Expert Systems with Applications, 2020, 150: 113302.

[23] DUARTE A, MARTI R, GLOVER F, *et al*. Hybrid scatter tabu search for unconstrained global optimization [J]. Annals of Operations Research, 2011, 183(1): 95.

[24] RUSSELL R A, CHIANG W C. Scatter search for the vehicle routing problem with time windows [J]. European Journal of Operational Research, 2006, 169(2): 606.

[25] GE J H, LIU X L, LIANG G. Research on vehicle routing problem with soft time windows based on hybrid tabu search and scatter search algorithm [J]. Computers, Materials & Continua, 2020, 64(3): 1945.

[26] SOLOMON M M. Algorithms for the vehicle routing and scheduling problems with time window constraints [J]. Operations Research, 1987, 35(2): 254.

[27] AWAD H, ELSHAER R. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants [J]. Computers & Industrial Engineering, 2019, 140: 106242.