

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

CrossMark 现实约束车辆路径问题的混合算法

张德福 a, 蔡 a, 芙蓉叶 a, j, 雅安-沃思 b, 中堂阮 c

中国厦门厦门大学信息科学与工程学院

一辆卡车

文章历史:

2016年2月7日收到

2017年2月10日修订

2017年2月14日接受 2017年2月15日

在线提供

关键词:

车辆路径问题

集装箱装载

禁忌搜索

人工蜂群算法

abstract

跨国公司的激增和极具竞争力的商业环境导致了对第三方物流服务前所未有的需求。然而,最近关于车辆路径问题(VRP)的研究只考虑了简单的限制。它们也不能很好地适应物流行业中遇到的现实问题。在本文中,我们引入了一个新的带有时间窗和托盘装载约束的车辆路径问题;这个问题解释了物流行业企业的实际需求,如消费品和农产品的运输。为了解决这个新的 VRP 问题,我们提出了一种结合禁忌搜索和人工蜂群算法的混合方法。生成新的基准数据集来验证所提出的算法的性能,因为所提出的 VRP 在文献中从未被报道过。对一组有时间窗的所罗门 56 个车辆路径问题进行了实验。我们的方法优于其他一些启发式算法在比较所罗门的 VRPTW 实例。

2017 爱思唯尔公司保留所有权利。

英国利物浦约翰穆雷斯大学海事与机械工程系

1. 介绍

车辆路径问题(VRP)是物流运输领域的经典问题。它关注的是从中央停车场出发前往一组客户的车辆的路线规划。由于其在商业和公共实体中的广泛应用,VRP 被认为是运筹学中最重要的问题之一[5]。VRP 最早是由丹齐格和拉姆齐提出的[11]。近年来,对这个问题的进一步改进和变化进行了广泛的研究。尽管近年来经典 VRP 引入了各种目标,但这些变体中的大多数大多与对原始问题添加新的约束有关。例如,托盘和时间窗口是 VRP 的各种研究人员最常使用的两个约束条件。在托盘约束下,不同尺寸的货物必须用标准尺寸和有限容量的箱子运输。在时间窗口约束下,货物必须在特定的时间窗口内交付。文献中报道了许多与上述限制相关的研究。对于托盘约束,梁等人。[25]利用模拟退火算法求解了具有二维装载约束的异构车队 VRPs。魏等。[39]提出了一种具有三维载荷约束的异构机群 VRP 的自适应可变邻域搜索。对于具有时间窗约束的 VRPs,Cherkesly 等人。[6]为有时间的提货和交货问题开发了分支降价算法。

*对应作者。

电子邮件地址:dfzhang@xmu.edu.cn。张(女。叶)。

窗口和多个堆栈。HiFi 和吴[17]通过依次考虑车辆数量和总距离, 解决了带有时间窗的问题。龚等。[15]开发了一种离散粒子群优化方法, 用于有时间窗的车辆路径问题。

然而, 在现有的研究中, 这两个限制通常被分开考虑。实际上, 这两种约束在运输问题中并存。如何解决日常配送问题是大多数物流公司普遍关注的首要问题。然而, 在设计简单的虚拟现实时, 研究人员经常忽略现实场景中遇到的一些限制。在大多数情况下, 这些约束是分开考虑的。因此, 包含单一约束的 VRP 通常有几个缺点。例如, 有时间窗的 VRP 侧重于建造路线, 而不考虑待运物品的属性。尽管有考虑车辆容量的带时间窗的虚拟现实, 但它们仍然忽略了物品的实际物理尺寸。另一方面, 有托盘约束的车辆路径规划只关注构建具有可行装载条件的路径, 而不考虑时间因素。由于忽略了时间因素, 对于一些现实场景, 货物可能无法按时交付给客户。因此, 有必要通过考虑时间窗口和托盘约束(尤其是三维装载问题)来进一步研究 VRP 的现实情况。

据我们所知, 具有时间窗和三维载荷约束的 VRP 只在扎卡里亚迪斯等人的工作中讨论过。[43]。[43]中定义的问题有特定的约束, 包括不同请求类型的混合。这个 VRP(在[43]中提出)可能无法满足某些物流公司的调度要求, 尤其是来自电子商务领域的物流公司。原因是, VRP 在[43]中提出的建议同时考虑了提货和送货活动, 而电子商务中的物流公司通常只涉及配送服务。

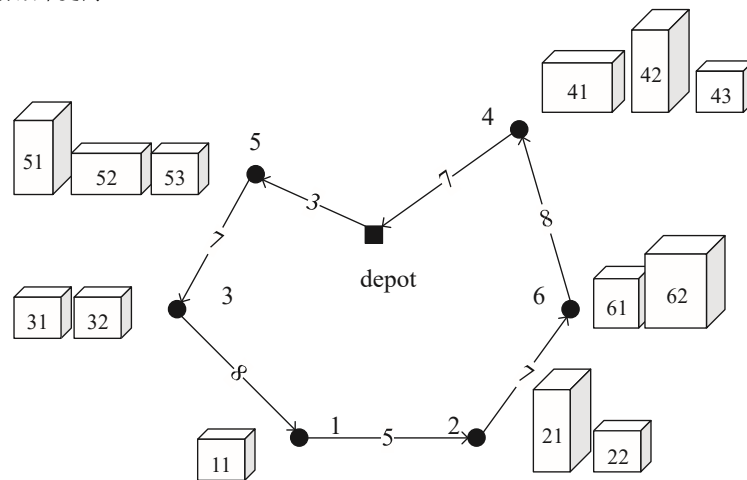
本文介绍的问题可以认为是一个 VRP 问题, 它包括两种类型的约束: 时间窗口和三维加载约束。与[43]相比, 本文中解决的问题更接近真实世界的情况, 因为它既考虑了业务的实际需求(详细描述如下), 也考虑了仓库和客户的独特类型。

对于第一种类型的约束, 仓库和客户被指定了一个时间窗口, 该时间窗口包括确切的开始和结束时间。这意味着车辆必须在车辆段的工作时间窗口内出发和返回。车辆还必须在规定的开始和结束时间之间拜访客户。如果车辆在客户开始时间之前到达, 任何卸货将被推迟, 直到客户有空。

除了基本的空间约束外, 还考虑了其他三维载荷约束, 如脆性、支撑面和卸载顺序。由于所提出的方法不仅考虑了时间窗口, 还考虑了来自现实场景的三维装载约束, 因此涉及电器运输和新鲜农产品配送的企业可以从所提出的解决方案中受益匪浅。对于只有装载约束的 VRP, 可以采用集装箱装载问题来验证装载的可行性。然而, 对于具有时间窗和托盘装载约束的 VRP, 装载问题可以被认为是一个三维箱柜装载问题, 其中固定数量的矩形物品被装载到较大的矩形箱子中[16]。

早期关于 VRP 的研究主要基于精确的方法。与此同时, 物流行业的快速发展和当今商业环境的动态性对虚拟现实平台的规模和复杂性产生了重大影响。显然, 传统的精确方法可能无法与现实世界的情况相匹配。这种情况导致了 VRP 启发式/元启发式算法的发展。禁忌搜索是早期启发式研究中最重要方法之一。关于使用禁忌搜索求解 VRP 问题的最早报告之一是由 Gendreau 等人提出的。[13]。此后, 许多研究人员使用禁忌搜索来求解虚拟现实问题。Taillard 等人使用了禁忌搜索和统一禁忌搜索。[34]和 Cordeau 等人。[9]解决具有软时间窗的 VRP 以及具有时间窗的 VRP。梁等。[26]将扩展的引导禁忌搜索和一种新的装箱算法应用于具有二维载荷约束的 VRP。Karaboga 在[20]中提出的人工蜂群算法是一种新的求解 NP-hard 问题的优秀算法。在过去的十年中, 基于人工蜂群算法的方法越来越多地被研究者用来解决虚拟现实问题。在[33]中, Szeto 等人应用人工蜂群算法求解了容量受限的 VRP 问题。后来, 姚等人。[40]提出了一种具有周期性 VRP 扫描策略的人工蜂群算法。同时, 其他启发式算法, 如模拟退火[21]和遗传算法[18], 也被广泛用于求解 VRPs。结合不同元试探法的混合方法也被用于解决虚拟现实问题; 例如, Bortfeldt 等人。[4]描述了一种用于 VRP 的混合算法, 该算法涉及聚集反向运输和装载约束。Beheshti 和 Hejazi [3]提出了一种列生成和元试探法的混合算法, 用于 VRP 的变体。张等。[45]通过将进化局部搜索与具有三维载荷约束的 VRP 的重组方法相结合, 开发了一种混合算法。Akpınar [1]针对容量受限的 VRP 提出了一种混合大邻域搜索算法。

我们工作的主要贡献是双重的。首先, 我们考虑 VRP 的一个新变体, 它涉及负载和时间窗口约束。这种变体旨在模仿物流行业的真实场景, 首次将两种约束结合起来。与以前的研究相反, 我们用一种有效的方法来解决负载约束。其次, 提出了一种结合禁忌搜索和人工蜂群算法的混合算法。在一组具有时间窗的所罗门 56 VRP(VRPTW)上, 对照其他启发式算法评估了该算法的性能。新的混合算法利用了两种启发式算法, 执行效率更高



每个顶点的时间窗口和工作时间([tsi, tei], ti): 仓库:[0, 70], 0

1:[15,23],2 2:[25,30],1 3:[10,14],1 4:[35,55],3
5:[2,8],2 6:[10,40],1

图。1.VRPTW 路由的示例。

为其他领域的问题提供了一种潜在的解决方法。此外，这项工作为虚拟现实技术工作组生成的一组基准数据可被其他相关方采用，以便在这一领域进行进一步研究。

本文的其余部分组织如下。第二节详细介绍了虚拟现实技术。在第三节中，我们详细描述了新的混合算法。第四节报道了算法的计算结果，最后总结了第五节的工作。

2. 问题是

设 $G(V, A)$ 为无向图，其中 $V = \{0, 1, \dots, n\}$ 是顶点集， A 是边集。设 C_{ij} 为顶点 i 和 j 之间的运输成本。 V 中的顶点 0 称为仓库。求解 VRP 的主要目标是寻找一个覆盖最短距离但遍历每个顶点的解。遍历必须在仓库开始和结束。

回想一下，本文要讨论的 VRP 既考虑了时间窗口，也考虑了负载限制，以反映真实世界的情况。在这种情况下，我们假设每个顶点都有一个客户，并且每个客户都需要仓库中的一组物品。每组物品都有一个预定义的总重量。每一项都可以认为是一个长 l_i ，宽 w_i ，高 h_i 的三维长方体。在车辆段顶点，有一些可运载货物的车辆，每个车辆都有一个尺寸为 $L \times W \times H$ 的固定装载空间(一个集装箱)，其中 L 、 W 和 H 分别是装载空间的长度、宽度和高度。此外，每辆车都规定了重量容量 D 。每个顶点可以是仓库，也可以是客户，每个顶点都有一个特定的工作时间窗口 $[t_{si}, t_{ei}]$ 和工作时间 t_i 。此外，特定顶点的所有工作必须在每个顶点的工作时间窗口内执行。图 2 给出了一个路由中时间窗口的例子。1.

在 VRPTWP 中，必须满足以下条件才能满足装载约束：

- (I) 每个客户(顶点)只被访问一次。换句话说，一个客户只属于一条路线。(二)所有路线必须在停车场开始和结束。
- (iii) 所有路线必须在车辆段的时间窗口内出发和返回。
- (iv) 所有客户(顶点)在其工作时间窗口内都是可用的。
- (v) 客户在路线上需要的所有物品都被装载到为该路线服务的车辆上。

在车辆能够离开仓库之前，有必要确保所有需要的物品都装载到车辆上。这可以被认为是验证三维加载问题可行性的过程，同时考虑以下条件：

- (一)所有物品必须完全装载并装入车辆的集装箱，即。物品和容器的边缘彼此不相交。(二)项目不允许重叠。
- (iii) 每个物品的底部必须由其他物品的底部或容器的底部充分支撑。
- (iv) 每个物品的表面必须与容器的表面平行。

除了上述基本条件之外，我们还添加了以下条件，以更好地反映现实世界的情况：

- (v) 方向: 项目具有固定的垂直方向，这意味着它们具有固定的底面。
- (vi) 能力: 项目重量的总和小于或等于车辆的装载能力。(七)易碎性: 非易碎物品不能放在易碎物品上。
- (八)后进先出法: 如果顾客 i 比顾客 j 早来，那么顾客 j 的商品应该比顾客 i 的商品早包装。

数学上，我们的目标是找到一个最小行驶距离，它可以表示如下：

$$\min \sum_{i=1}^r \sum_{j=1}^{n_i} C_{ij} \quad (1)$$

其中 r 为路线数(车辆数等于路线数，因为每辆车都服务于唯一的路线)， n_i 为路线 i 中的顶点数， C_{ij} 表示从顶点 j 到顶点 $j+1$ 的行驶成本(本文中，行驶成本等于距离)。

时间窗口约束可以描述如下：

$$t_{ai} < t_{ei} \quad (2)$$

在哪里

$$t_{ai} = t_{l(i-1)} + t_{C(i-1)}, \quad (3)$$

如果 $t_{ai} > t_{si}$ ，则 $t_{ai} + t_i$

$$t_{li} = (4)$$

$t_{si} + t_i$, 否则

其中 t_{si} 和 t_{ei} 是路径中顶点 I 的开始和结束时间, t_{ai} 是车辆到达顶点 I 的时间, t_{li} 是车辆从顶点 I 离开的时间, t_i 是在路径中为顶点 I 工作的时间成本, $t_{CI}(I-1)$ 是从顶点 I 行驶到顶点 $I-1$ 的时间成本。

因为装载约束满足许多条件, 并且用启发式算法处理, 所以不需要提供该约束的数学公式。请注意, VRPTWP 可以被认为是一个 NP 难问题的组合。因此, VRPTWP 可以归类为一个更难的 NP-hard 问题。在下一节中, 我们将描述一个启发式算法来解决这个问题。

3. 提议的方法

根据第 2 节中给出的描述, VRPTWP 可以被视为具有三维载荷和时间窗约束的 VRP 的组合。在本节中, 提出了一种两阶段方法。首先, 我们考虑多种策略来解决三维加载问题。其次, 结合禁忌搜索和人工蜂群方法, 采用混合算法求解 VRPTW 问题。

3.1. 三维载荷问题

提议的方法的一个关键方面是使用一种创新的方法来判断沿途客户需要的箱子(物品)是否可以装载到车辆上。这种可行性测试对结果有显著影响, 因为它被第 3.2 节中描述的主要算法反复调用。

3.1.1. 装载位置

(车辆的)容器放置在三维坐标系统中(见图. 2.), 并且坐标的原点位于容器的左下角。 l 、 W 和 H 代表容器的长度、宽度和高度。为了充分利用可用空间, 要装载的箱子需要靠近集装箱或已经装载的箱子。} or {因为箱子可以在水平方向上旋转, 当长度为 l_i 、宽度为 l_i 、高度为 l_i 的箱子被装入集装箱时, 其沿 x 、 y 和 z 轴的值可以被认为是 l_i 、 l_i 、 l_i 。

我们表示 $B = \{b_1, b_2, \dots, b_n\}$ 作为一套盒子。首先, 如果在 $(0, 0, 0)$ 有可用的加载位置, b_1 将被加载。接下来, b_2 有三个可用的加载位置, 即 $(l_1', 0, 0)$, $(0, w_1', 0)$ 和 $(0, 0, h_1')$ 。假设 b_2 加载在位置 $(l_1', 0, 0)$; 然后, $(l_1', 0, 0)$ 被删除, 另外三个可用的装载位置 $(l_1' + l_2', 0, 0)$, $(l_1', w_2', 0)$ 和 $(l_1', 0, h_2')$ 将被生成。因此, b_3 现在有五个装载位置。一般情况下, 如果第 I 个箱子在位置 (x, y, z) 装载, 那么 (x, y, z) 将从可用装载位置列表中删除。随后, $(x + l_i', y, z)$, $(x, y + w_i', z)$ 和 $(x, y, z + h_i')$ 将被添加到可用装载位置列表中(见图. 1.). 当一个盒子被装载时,

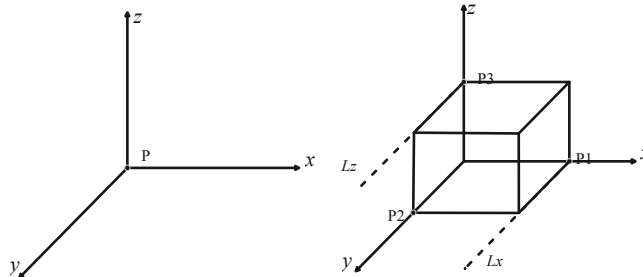


Fig. 2. Updating process of loading positions.

Algorithm1 Verifying the feasibility of three-dimensional loading.

PalletisationFeasibility(Vehicle)

1. $I = \{(0, 0, 0)\}, L_z = L_x = 0;$
2. **对于** $i = 0$ **至** n
3. $flag = false$
4. **对于** $(x, y, z) \in I$
5. **如果** $x + l_i' \leq L_x, z + h_i' \leq L_z$ **和** b_i **我可以被载入** (x, y, z)
6. $flag = true$, **转到**第 19 行;
7. **如果** $L_x = 0$ **或** $L_x = 1$
8. **如果** b_i **可以加载在** $(0, 0, L_z)$ **上**
9. $x = 0, y = 0$, **标志=真**, $z = L_z, L_z = L_z + h_i', L_x = l_i$
10. **其他**
11. $L_z = H, L_x = L, I = I - I;$
12. **其他**
13. **对于** $(x, y, z) \in I$

```

14.      如果  $x = Lx$ ,  $y = 0$ 
15.      如果  $z$  是  $+h_i' \leq Lz$  和  $b$ , 我可以加载  $(x, y, z)$ 
16.       $Lx = Lx + l_i'$ , 标志 = true, 转到第 19 行;
17.      其他
18.       $Lx = L$ ,  $I = I - 1$ ;
19.      如果标志 = 真
20.      在  $(x, y, z)$  上加载  $B1$ ,  $I = I / (x, y, z)$ , 平移移动  $B1$ 
      运算符并标记  $b_i$  的新位置  $(x', y', z')$ ,  $I = I \cap (x' + l_i', y', z')$ ,  $(x', y' + w_i', z')$ ,
       $(x', y' + w_i', z')$ ;
21.      其他
22.      返回 false
23.      返回真;

```

将删除一个可用装载位置，并添加三个新的可用装载位置。因此， b 箱 I 将有 $2(i-1) + 1$ 个可用装载位置。如果 b_1 不能装载到任何可用的装载位置，那么假设箱子 $\{b_1, b_2, \dots, b_n\}$ 不能装在容器中。

3.1.2. 参考线

本研究中考虑了两条参考线来控制已用空间和有效包装。 Lz 和 Lx 分别表示 z 轴和 x 轴的两条参考线。当检查是否可以在位置 (x, y, z) 装载箱 b_i 时，我们必须确保不违反第 2 节中规定的条件。此外，加载还必须满足 $z + h_i < Lz$ 和 $x + l_i < Lx$ 的条件。一旦一个可用装载位置被认为是可行的， b_i 将被装载，并且可用装载位置列表将被更新。当没有可用的位置时，考虑以下两种情况：(1) 如果 $Lx < L$, Lx 将增加到 L ; (2) 如果 $Lz < H$, Lz 将增加到 H 。如果在 (1) 和 (2) 之后没有找到 b_i 的可行位置，那么我们可以得出结论，箱子不能装入集装箱。

3.1.3. 平移算子

一旦选择了可用的装载位置并装载了箱子，箱子的位置将通过减少 x 、 y 和 z 的值来调整，直到它被其他箱子或容器阻挡。

3.1.4. 加载算法

在本文中，我们采用了 [44] 中的加载算法 (算法 1)。 I 是可用装载位置的列表，根据 x 按递增顺序排序。当加载位置的 x 值相同时，参数 y 和 z 用于断开连接。当列表 I 被更新时，我们保持它的顺序。此外，一个标志变量用来表示一个盒子是否可以被装入容器。装载算法接受订购的箱子 B 和车辆的集装箱作为输入。如果所有的盒子都可以装入容器，算法将返回真。

Algorithm2 Initialization of VRPTW.

初始化(C)

```

1.      Route_Num = 0,  $c = C$ ,  $S = \phi$ ;
2.      而  $\neq \phi$ 
3.      从  $C$  中随机选择客户  $c_i$ 。
4.      如果路由号 = 0
5.      Route_num = Route_num + 1;
6.      生成新的路由  $S$ ,  $S = S + S$ ;
7.      插入  $c_i$  到  $s$ ,  $c = c/c_i$ ;
8.      否则, 如果  $c_i$  可以插入到  $S$  中的路由中
9.      选择插入后成本增加最少的;
10.     在插入后成本增加最少的位置将  $c_i$  插入  $s$ ;
11.      $c = c/c_i$ ;
12.     其他
13.     Route_num = Route_num + 1;
14.     如果路线编号 > 车辆数量
15.     转到第 1 行;
16.     其他
17.     生成新的路由  $S$ ,  $S = S + S$ ;
18.

```

入 c_i 到 s , $c = c/c_i$;

c 是客户集, S 是路由集, $Route_Num$ 是路由数。

插

首先, 原始可用装载位置设置为(0, 0, 0), L_z 和 L_x 初始化为 0。箱子已经准备好了, 可以按顺序装入集装箱。在这个算法中, 首先测试每个盒子, 看它是否能被装入容器而不超过参考线。如果无法加载, 则算法会更改 L_x 的值。当 L_x 等于 0 或 L 时, 算法增加 L_z 。当 L_z 等于 H 时, 如果盒子仍然无法加载, 算法增加 L_x 。当算法试图增加 L_x 的值时, 所有满足 $x = L_x$ 和 $y = 0$ 的位置都要测试装入盒子的能力。一旦 L_x 等于 L 并且无法加载该框, 算法将返回 false。当一个箱子装载成功时, 选定的装载位置从列表 I 中删除。随着位置的调整移动箱子后, 会生成三个新的可用装载位置。如果所有的框都成功加载, 算法将返回 true 否则, 它将返回 false。

3.2. VRPTW 解决方案的总体结构

回想一下, VRPTW 由两个问题组成, 即具有三维载荷约束的 VRP(3L-CVRP)和 VRPTW。因为前一个问题已经在 3.1 节中解决了, 我们现在准备解决 VRPTW。在我们的方法中, 我们使用局部搜索来改进解决方案。我们定义了六个邻域结构。在每次迭代中随机选择其中一个, 算法试图在每次迭代中找到一个成本更低的更好的解。然而, 这种策略会导致算法陷入局部最优。为了避免局部最优, 我们采用了禁忌搜索和人工蜂群算法。

3.2.1. 构建最初的 VRPTW 解决方案

我们的算法首先生成一个初始可行解(解释如下)。一旦这个初始化步骤完成, 就可以进行进一步的改进以获得更好的结果。请注意, 第 2 节中描述的所有约束都必须由这个初始解来满足。

在该算法中, 客户被一个接一个地随机插入到路线中。在插入过程中, 客户被插入成本增加最小的位置, 前提是满足第 2 节中提到的所有限制。插入新客户后, 如果不能为该客户分配所有现有路线, 将建立新路线。如果路线总数超过可用车辆数, 算法将重新启动。否则, 算法将继续, 直到处理完所有客户。算法 2 描述了构建过程。

3.2.2. 邻域解决方案

当我们试图找到一个更好的解决方案时, 我们可以沿着不同的路线重新分配顶点的位置来构造一个新的解决方案。在每一步, 我们随机选择一个邻域结构。本文应用了六种邻域结构, 定义如下:

- 交换: 在这个策略中, 两个客户的位置被交换。两个客户可以在同一条路线上(见图. 3 (a)). 或者在不同的路线上(见图. 3 (b)).
 - 重新定位: 通过这种策略, 客户被转移到另一个位置。新位置可以在其原始路线上(见图. 3 (c)). 或者在不同路线上(见图. 3 (d)).
 - 路线交换: 每个路线被一个顶点(客户)分成两个子路线。两条路线的两条子路线(特定顶点后访问的部分)完全交换(见图. 3 (f)).
- (四)路线反转: 子路线反转客户的顺序(见图. 3 (e)).

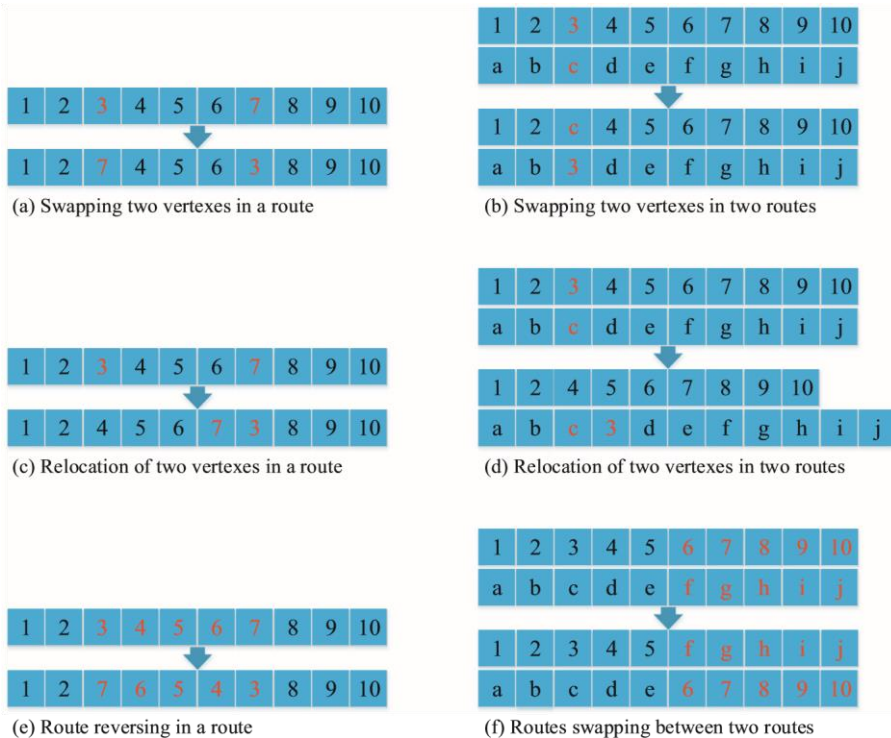


图. 3. 邻域结构。

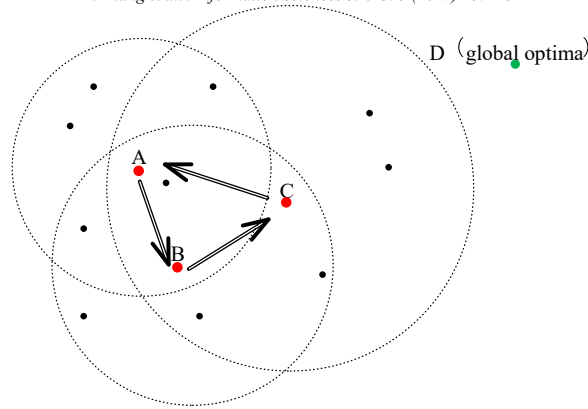


Fig. 4. Trap of local optima.

3.2.3. 禁忌语

禁忌搜索算法是禁忌搜索和人工蜂群算法的结合。它使用禁忌搜索来快速生成高质量的解，该解被人工蜂群算法使用。同时，人工蜂群算法利用禁忌搜索增加食物来源的多样性。

禁忌搜索旨在搜索其邻域内的最佳解，即使没有更好的解。这种方法有陷入局部最优的风险(图. 4)。a、B、C 和 D 是四种不同的解。在 A 的邻居中，B 是最好的解决方案。C 是 B 邻域的最优解，A 是 C 邻域的最优解。根据该算法，找到的最佳解只能是 B、C 和 a。在这种情况下，局部搜索陷入循环，解决方案无法进一步改进。虽然 D 是真正的最优解，但是被算法错过了。这种情况突出了搜索陷入局部最优的问题。

为了避免这个陷阱，引入了一个禁忌列表来禁止算法重新访问以前访问过的客户。一旦我们找到了更好的邻居结构解决方案，所有最近交换的客户都会被插入到禁忌列表中。在随后的搜索迭代中，这些客户将不会被选中，除非算法可以

算法 3 禁忌搜索算法的框架。

禁忌语

1. 找到原始食物来源 x_i , $i=1, 2, \dots, n$, 带初始化; 2. 评估每种食物来源的适合度 $f(x_i)$, $i=1, 2, \dots, n$;
3. 用禁忌搜索提高食物来源的适合度(在附近搜索新的食物来源, 如果一个新的食物来源比旧的好并且不在禁忌任期内, 用它替换旧的);
4. 无更新次数: $l_1=l_2=\dots=l_n=0$;
5. **而不满足终止条件 do**
6. 就业蜜蜂阶段: 根据 x_i 的传统在附近搜索 x_i , 如果轮盘赌 $f(x_i')$ 是轮更大的选择; 比 $f(x_i)$, 求用 x_i in x_i 替换 $bestx_i$, 邻域 $l_i=0$; 否则 $l_i=l_i+1$ 的, if l_i 更大
7. 旁观者阶段: 选择一个 x_i
- 比 $f(x_i)$, 用 x_i 代替 x_i , $l_i=0$; 否则 $l_i=l_i+1$;
8. 侦察蜂阶段: 对于每个食物源 x_i , 如果 $l_i=limit$, 则找到一个与所有已有食物源相似度低于预定限制的新食物源, 并用其替换 x_i ;
9. **结束时间。**

获得比当前最佳解决方案更好的解决方案。禁忌列表的最大长度称为禁忌任期。

人工蜂群算法是一类群体智能技术。算法中的蜜蜂分为三类: 雇佣蜂、旁观者和侦察兵。雇佣蜜蜂的工作是开发食物来源。他们收集信息并与旁观者分享。根据被雇佣的蜜蜂分享的信息, 旁观者选择质量更高的食物来源。因此, 高质量的食物来源将由在线观看者选择。侦察蜂随机探索新的食物来源。当旁观者和侦察蜂找到新的食物来源时, 它们就变成了雇佣蜂。

由于禁忌搜索从随机解开始, 其性能很容易受到初始解的影响。因此, 我们引入人工蜂群算法来缓解这一缺点。人工蜂群算法通过消除初始解的影响, 生成一组禁忌搜索的初始解。由于禁忌搜索可以帮助人工蜂群产生更好的食物来源, 因此提出了一种混合算法。

禁忌-作业成本法(算法 3)首先生成随机解作为食物来源, 并将每个来源与雇佣的蜜蜂联系起来。在这个关联事件之前, 应用禁忌搜索来改进初始解(食物来源)。然后, 每只被雇佣的蜜蜂确定其相关食物来源附近的新食物来源。如果它找到了新的更好的食物来源, 它会从旧的来源切换到新的来源。在所有被雇佣的蜜蜂完成他们的工作后, 这发生在一个固定的迭代中, 他们与旁观者分享他们的信息。旁观者然后根据传统的轮盘赌选择方法选择食物来源。在旁观者选择了他们的食物来源之后, 他

们探索并评估离所选食物来源很近的食物来源。对于每一个旧的食物来源，如果找到了新的更好的食物来源，旧的来源就会被新的来源所取代。此外，如果一个食物源在预定的迭代次数中没有表现出改善，那么它就被放弃了。在那时，被雇佣的蜜蜂被转化为侦察兵，并将与一个新的食物来源相关联。在这里，我们再次应用禁忌方法。新食物来源和旧食物来源的相似性不能超过预定的阈值。新的食物来源被随机生成，直到其中一个满足相似性条件。在本文所述的 VRPTW 的上下文中，路由 a 与路由 b 的相似性定义为它们最长的公共子序列的长度除以路由 a 的长度。解决方案 A 与解决方案 B 的相似度是 A 的每条路线与 B 的路线的最高相似度的平均值。当算法达到预定的迭代次数时，算法终止。

4. 计算结果

为了获得 VRPTW 的有效解决方案，并为进一步的研究提供比较基础，我们为 VRPTW 构建了一组基准，并报告了它们的计算结果。为了生成基准，我们结合了两个著名的实例。第一个是[14]中为 3L-CVRP 提出的一组例子，另一个是所罗门[32]介绍的 VRPTW 的著名例子。3L-CVRP 实例中的位置信息被所罗门的 27 个实例(C1、C2 和 R101 - R110)逐个替换。同时，时间窗口信息也被导入。为了保持实例的可行性，允许的最大车辆数量是两倍。我们生成了一组基准并测试了建议的方法。我们相信，所获得的计算结果可以用作该领域未来发展的基线方法。

在这一部分中，给出了虚拟现实技术工作组和 VRPTW 的计算结果。所提出的算法是用 C++ 编码的，运行在一台配备英特尔酷睿 i5 处理器、2.6 千兆赫/8 G 内存的机器上。

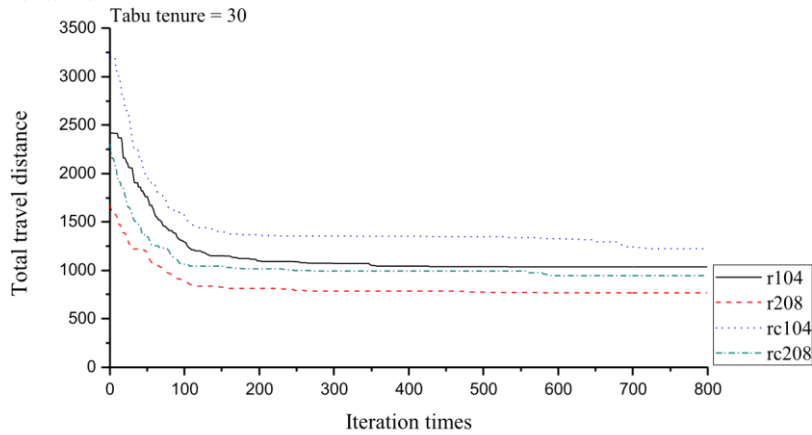


Fig. 5. Process of Tabu search on some instances.

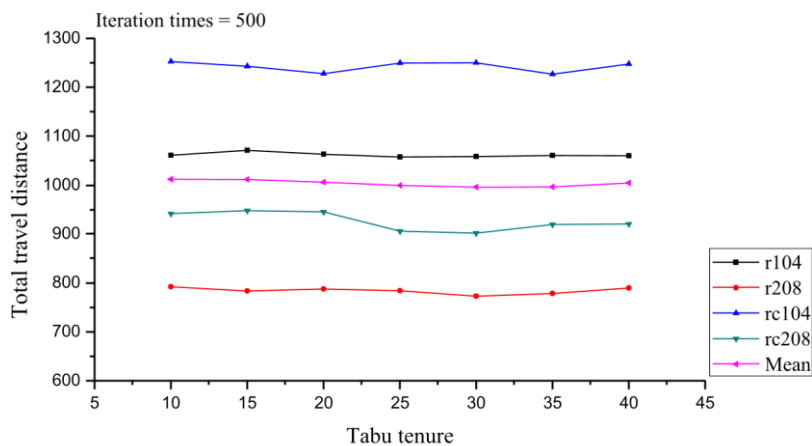


Fig. 6. Results on different Tabu tenure.

4.1. 参数敏感性分析

为了分析参数的影响，我们进行了许多不同设置的实验。为了找到时间消耗和解决方案质量之间的关系，我们在四个实例上应用了禁忌搜索，在 10 次运行中进行了 800 次迭代。图 5 展示了实例 r104、r208、rc104 和 rc208 上的 Tabu 搜索过程。横轴和纵轴表示迭代次数和总行程距离(解决方案的成本)。图 5 显示，在 300 次迭代后，实现的改进非常小。每个实例的最低总行驶距离分别为 1037.95、758.38、1224.74 和 946.74。在图 1 中。图 6 显示了不同 Tabu 任期的平均结果。显然，在大多数情况下，拥有 30 个 Tabu 任期的旅行距离比拥有其他 Tabu 任期的旅行距离短。为了测试禁忌搜索的更多参数。图 7 显示了更新限制的影响(没有更新的最大允许迭代次数)。8 表示相似性限制的影响(允许侦察蜂接受新来源的最小相似性)。根据图 7、更新限制和相似性限制对最终结果影响不大。尽管更新限制的不同值可能会对特定实例(如 r104 和 rc208)产生很大影响，但它们平均仍显示出相似的结果。当相似度限制为 0.7 时，四个实例的平均结果最短，结果与其他集合相似。事实上，从图中我们可以发现，Tabu-ABC 是稳健的，参数对总行程的影响较小。对于时间成本，对于 VRPTWP 实例上的每个测试，Tabu 保有权的长度被设置为 30，并且 Tabu 搜索被进行 300 次迭代。食物来源的组大小也设置为 10，不更新迭代次数的限制设置为 10 n (n 为顾客数)。源相似性的限制被设置为 0.7，并且 ABC 被进行 50 n 次迭代。对于 VRPTW 上的每一次测试，禁忌搜索都要进行 500 次迭代，而食物来源的种群规模被设置为 20。ABC 进行了 200 n 次迭代，其他参数的设置与 VRPTWP 实例上的设置相同。参数设置如表 1 所示。

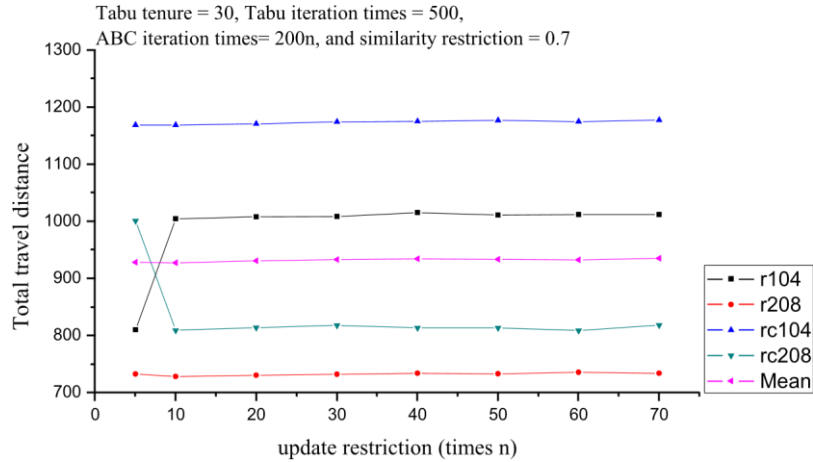


Fig. 7. Results on different update restriction.

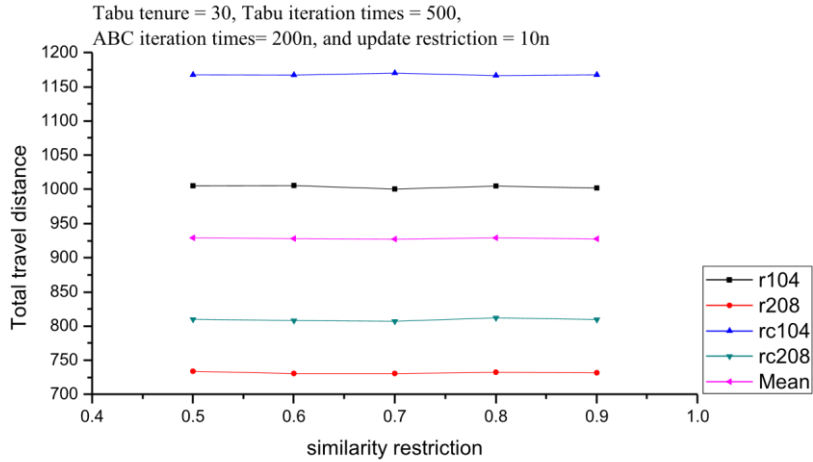


Fig. 8. Results on different similarity restriction.

表 1
参数设置。

禁忌保有权	30
禁忌搜索迭代次数	VRPTWP 300 人，VRPTW 500 人
食物来源的群体规模	10 人在 VRPTWP，20 人在 VRPTW
无更新次数限制	10 n，n = 客户数量
资源相似性的极限	0.7
作业成本法迭代次数	VRPTWP 为 50 牛顿，VRPTW 为 200 牛顿

4.2. VRPTWP 实例的计算结果

VRPTWP 的结果如表 2 所示，其中“NV”表示车辆数量，“TD”表示总行驶距离(解决方案成本)，“CPU”表示计算时间(秒)。算法为每个实例分配了 10 次独立运行。所提出的禁忌搜索算法计算量大，因为它解决了三维加载问题。文献[44]给出了三维载荷问题的一些计算结果。根据表 2，更大的车队规模不一定会导致更高的成本。正如在 VRPTWP15 和 VRPTWP22 中看到的，与所有运行中的平均车队规模相比，最低成本实际上会导致更大的车队规模。类似的情况见表 5 和表 6。事实上，从长远来看，购买更多车辆的成本远远低于旅行更远距离的成本。

4.3. VRPTW 实例的计算结果

我们工作的主要贡献包括新的 VRPTWP 问题和一种称为禁忌-作业成本法的新策略。为了评估算法的性能，我们在所罗门的 100 个客户上进行了广泛的实验

Table 2
Results of the VRPTWP.

Data set	Best		Mean		Gap	Sd	CPU
	NV	TD	NV	TD			
VRPTWP01	5	322.33	5	322.33	0	0	325.10
VRPTWP02	5	295.29	5	299.89	0.02	3.61	232.28
VRPTWP03	5	303.60	5	303.60	0	0	430.27
VRPTWP04	6	380.19	6	380.80	0	0.57	371.88
VRPTWP05	七	416.35	七	417.66	0	2.95	545.72
VRPTWP06	七	408.99	七	409.09	0	0.07	306.38
VRPTWP07	七	407.91	七	409.89	0	1.81	604.70
VRPTWP08	七	425.90	8	425.90	0	0	637.32
VRPTWP09	8	530.50	10	532.63	0	1.80	549.60
VRPTWP10	10	668.74	11	669.11	0	0.83	956.40
VRPTWP11	11	619.34	9	626.27	0.01	4.21	1032.98
VRPTWP12	9	688.60	10.40	690.10	0	1.56	671.37
VRPTWP13	11	626.72	8.40	630.43	0.01	4.40	1347.24
VRPTWP14	8	765.37	12	766.78	0	1.93	1221.68
VRPTWP15	12	713.23	10.60	715.72	0	3.41	1091.86
VRPTWP16	11	746.67	11.80	748.15	0	0.83	429.32
VRPTWP17	15	994.28	15.20	997.27	0	4.55	484.99
VRPTWP18	18	1206.51	18	1207.60	0	1.49	636.18
VRPTWP19	16	1211.99	15.60	1217.63	0	4.87	783.59
VRPTWP20	24	1644.56	23.20	1652.96	0.01	6.52	1512.11
VRPTWP21	23	1603.88	22	1612.42	0.01	9.29	2174.74
VRPTWP22	26	1811.19	26.20	1817.75	0	5.83	2170.98
VRPTWP23	24	1654.13	24	1675.25	0.01	15.22	1950.79
VRPTWP24	21	1644.55	21.40	1649.61	0	7.22	1745.54
VRPTWP25	27	1836.02	26.80	1856.52	0.01	16.43	2877.48
VRPTWP26	32	2144.47	32	2160.37	0.01	10.42	3250.49
VRPTWP27	28	2002.63	29.60	2030.71	0.01	21.35	3037.26

间隙=(平均 TD-最佳 TD)/最佳 TD，标清为 15 倍 TD 的标准偏差。

VRPTW，并将结果与其他启发式方法进行了比较。所罗门的 VRPTW 被分为六组，C1、C2、R1、R2、RC1 和 RC2.C1 和 C2 的顾客是成群结队的。在 R1 和 R2 的集合中，它们是均匀分布的，而在 RC1 和 RC2 的集合中，它们是半聚集的。所提出的算法在每个实例上执行了 10 次独立运行。

为了证明其有效性，将禁忌-作业成本法与其他九种启发式方法进行了比较。表 3 给出了比较结果，其中显示了每组的平均最佳总距离和车辆数量。显然，该算法在 C1、R2 和 RC2 取得了最好的结果，平均结果也优于其他一些启发式算法。在表 4 中，我们比较了每组的平均总距离和车辆数量。我们的算法仍然达到四个集合(R1、R2、RC1 和 RC2)的最佳平均解。

在表 5 中，我们将禁忌搜索算法与最近发表的一些关于 VRPTW 的论文中的启发式算法进行了比较。从实验结果中，我们可以观察到，在大多数情况下，所提出的算法获得了最佳解。我们还将我们的工作与表 6 中列出的最知名的解决方案进行了比较。大多数最著名的结果总结在[42]中。其中一些结果是根据我们最近的发现更新的。从表 6 中，我们可以观察到我们的算法在 15 个实例中获得了更好的解决方案。在 4 个实例中，它实现了最著名的解决方案。在其他情况下，所提出算法的性能也与最著名的解决方案相似。此外，Tabu-ABC 在图 2 中的四个具体实例上取得了较好的结果。5，体现了 Tabu-ABC 对纯 Tabu 的有效改进。因为表 6 中的 CPU 时间很大，所以我们用不同的参数来测试算法。从表 7 可以看出，该算法以较少的 CPU 时间获得了良好的结果。结果还表明，参数值略微影响禁忌-作业成本法的性能。

5. 结论

介绍了一种新的带时间窗和托盘装载约束的车辆路径问题。VRPTWP 由两个亚 NP 难问题组成，即三维载荷问题和 VRPTW 问题。此外，本文提出的 VRPTWP 考虑了时间窗口的限制，并紧密反映了现实世界的情况。据我们所知，VRPTWP 是一个从未解决过的新问题。为了寻找最优解，本文提出了一种新的算法——禁忌搜索算法。禁忌-作业成本法是一种基于禁忌搜索和作业成本法的混合算法。我们还为 VRPTWP 创建了一组新的基准。实验结果表明，与其他启发式算法相比，该方法在所罗门 VRPTW 问题上是非常有效的。对于未来的工作，我们希望开发更有效的方法来解决虚拟现实工作组，特别是在提高禁忌-作业成本法的中央处理器时间成本方面。因为 Tabu-ABC 可以很容易地针对不同的需求进行修改和扩展，所以我们计划在其他问题领域使用这种方法。

感谢

作者要感谢匿名推荐人的宝贵意见，帮助改进了这篇论文。本工作得到了国家自然科学基金(批准号:)资助的课题。61672439 和 61272003)，并由澳门大学在 MYRG2015-0 0 054-FST 的资助下提供，并且还部分得到英国特许物流和运输学会颁发的种子玉米资助和英国文化协会的两项资助、英国-东盟知识部分资助和牛顿机构链接资助。

附录

(表 3-7)。

表 3
VRPTW 上不同启发式算法的比较(最优解的平均值)。

日期设置	谭等(2006) [35]	于等(2011) [42]	Cordeau 和 Maischberger (2012) [10]	龚等(2012) [15]
C1	内华达州	10	10	10
	推倒 (网络用词)	828.71	828.38	835.91
C2	内华达州	3	3	3
	推倒 (网络用词)	590.07	589.86	593.41
R1	内华达州	12.92	13.1	12.58
	推倒 (网络用词)	1187.35	1209.19	1232.28
R2	内华达州	3.55	4.6	3
	推倒 (网络用词)	951.74	951.17	1016.66
RC1	内华达州	12.38	11.5	12.13
	推倒 (网络用词)	1355.37	1385.9	1385.47
RC2	内华达州	4.25	3.25	3.38
	推倒 (网络用词)	1068.26	1120.53	1169.07
平均值。	内华达州	6.59	6.07	6.30
	推倒 (网络用词)	996.92	1014.17	1038.80

	络用 词)	巴布查。(2014) [2]	罗等(2015) [28]	Yassen 等人(2015 年)[41]	禁忌语
C1	内华	10	10	—	10
	达州 推倒 (网 络用 词)	828.38	828.38	838.47	828.38
C2	内华	3	3	—	3
	达州 推倒 (网 络用 词)	589.86	589.86	605.41	590.39
R1	内华	11.92	11.92	—	13.75
	达州 推倒 (网 络用 词)	1232.13	1210.34	1207.76	1187.90
R2	内华	3.09	2.73	—	4.64
	达州 推倒 (网 络用 词)	922.48	951.03	977.19	891.24
RC1	内华	12	11.5	—	13.13
	达州 推倒 (网 络用 词)	1355.36	1384.16	1381.96	1361.08
RC2	内华	3.38	3.25	—	5.5
	达州 推倒 (网 络用 词)	1106	1119.24	1099.12	1017.47
平均值。	内华	7.23	7.07	—	8.34
	达州 推倒 (网 络用 词)	1005.70	1013.84	1018.32	979.41

表 4
VRPTW(平均解的平均值)上不同试探法的比较。

		日期设置				蒋和
罗素(1997) [7]		Lau 等人。(2003) [24]		谭等。(2006) [35]于等。(2011) [42]		
C1	内华	10	10	—	10	
	达州 推倒 (网 络用 词)	828.38	828.38	837.21	841.92	
C2	内华	3	3	—	3.3	
	达州 推倒 (网 络用 词)	591.42	589.86	632.42	612.75	
R1	内华	12.17	12	—	13.1	
	达州 推倒 (网	1204.19	1217.73	1240.31	1213.16	

	络用 词)				
R2	内华 达州 推倒 (网 络用 词)	2.73 986.32	2.73 967.75	— 1068.57	4.6 952.3
RC1	内华 达州 推倒 (网 络用 词)	11.88 1397.44	11.63 1382.42	— 1381.23	12.7 1415.62
RC2	内华 达州 推倒 (网 络用 词)	3.25 1229.54	3.25 1129.19	— 1154.88	5.6 1120.37
平均值。	内华 达州 推倒 (网 络用 词)	7.17 1039.55	7.10 1019.22	— 1052.44	8.22 1026.02
		Cordeau 和 Maischberger (2012) [10]	龚等(2012) [15]	罗等(2015) [28]	禁忌语
C1	内华 达州 推倒 (网 络用 词)	10 828.94	10 856.44	10 828.38	10 828.73

(下一页继续)

表 4(续)

日期设置		蒋和罗素(1997) [7]	Lau 等人(2003 年)[24]	谭等(2006) [35]	于等(2011) [42]
C2	内华达州推倒（网络用词）	3 590.85	3.03 612.93	3 589.86	3 591.45
R1	内华达州推倒（网络用词）	12.02 1213.57	13.01 1263.25	11.92 1210.75	13.82 1195.49
R2	内华达州推倒（网络用词）	2.73 959.62	3.1 1073.72	2.7 951.51	4.5 902.88
RC1	内华达州推倒（网络用词）	11.55 1386.39	12.66 1400.97	11.5 1384.62	13.56 1373.25
RC2	内华达州推倒（网络用词）	3.25 1130.27	3.59 1228.95	3.25 1119.63	5.47 1028.92
平均值。	内华达州推倒（网络用词）	7.09 1018.27	7.57 1072.71	7.06 1014.13	8.39 986.79

表 5
四种启发式方法的比较。

算法								
HSFLA(2015)[28]CPLA(2014)[2]PITSH(2012)[10]Tabu-ABC								
数据集	推倒（网络 用词）	内华达州	推倒（网络 用词）	内华达州	推倒（网络 用词）	内华达州	推倒（网络 用词）	内 华 达 州
C101	828.94	10	828.94	10	828.94	10	828.94	10
C102	828.94	10	828.94	10	828.94	10	828.94	10
C103	828.06	10	828.06	10	828.07	10	828.07	10
C104	824.78	10	824.78	10	824.78	10	824.78	10
C105	828.94	10	828.94	10	828.94	10	828.94	10
C106	828.94	10	828.94	10	828.94	10	828.94	10
C107	828.94	10	828.94	10	828.94	10	828.94	10
C108	828.94	10	828.94	10	828.94	10	828.94	10
C109	828.94	10	828.94	10	828.94	10	828.94	10
C201	591.56	3	591.56	3	591.56	3	591.56	3
C202	591.56	3	591.56	3	591.56	3	591.56	3
C203	591.17	3	591.17	3	591.17	3	591.17	3
C204	590.6	3	590.6	3	590.6	3	594.89	3
C205	588.88	3	588.88	3	588.88	3	588.88	3
C206	588.49	3	588.49	3	588.49	3	588.49	3
C207	588.29	3	588.29	3	588.29	3	588.29	3
C208	588.32	3	588.32	3	588.32	3	588.32	3
R101	1650.8	10	1656.21	19	1650.8	19	1643.18	20
R102	1486.12	17	1501.97	17	1486.12	17	1460.26	18
R103	1292.67	13	1295.6	13	1294.23	13	1217.39	15
R104	1007.31	9	1017.38	9	981.2	10	987.61	11
R105	1377.11	14	1381.89	14	1377.11	14	1363.91	15
R106	1252.03	12	1258.76	12	1252.62	12	1247.90	13
R107	1104.66	10	1117.85	10	1104.66	10	1087.50	12
R108	960.88	9	976.06	9	963.99	9	961.85	11
R109	1194.73	11	1229.71	11	1194.73	11	1152.99	13
R110	1118.84	10	1196.49	10	1118.84	10	1091.50	12
R111	1096.73	10	1123.64	10	1096.73	10	1067.46	12
R112	982.14	9	1030.02	9	989.27	9	973.25	10
R201	1252.37	四	1253.02	四	1252.37	四	1174.69	6
R202	1191.7	3	1086.08	四	1191.7	3	1046.10	5
R203	939.5	3	945.8	3	941.08	3	884.02	5
R204	825.52	2	752.13	3	825.52	2	750.40	四
R205	994.43	3	1017.93	3	994.43	3	960.75	5
R206	906.14	3	920.37	3	906.14	3	900.97	四
R207	890.61	2	815.26	3	890.61	2	809.72	四
R208	726.82	2	729.42	2	726.82	2	723.14	5
R209	909.16	3	916.33	3	909.16	3	863.12	5
R210	939.37	3	943.1	3	939.37	3	927.54	5
R211	885.71	2	767.82	3	885.71	2	763.22	四
RC101	1696.95	14	1626.09	15	1696.95	14	1646.17	16
RC102	1554.75	12	1486.17	13	1554.75	12	1481.61	14
RC103	1261.67	11	1268.79	11	1261.67	11	1280.76	12
RC104	1135.48	10	1136.27	10	1135.48	10	1162.03	11
RC105	1629.44	13	1542.29	14	1633.72	13	1545.30	16
RC106	1424.73	11	1394.1	12	1424.73	11	1401.17	14
RC107	1230.48	11	1234.06	11	1232.2	11	1235.28	12
RC108	1139.82	10	1155.1	10	1147.69	10	1136.35	11
RC201	1406.94	四	1435.27	四	1406.94	四	1271.78	七
RC202	1365.64	3	1162.8	四	1367.09	3	1116.21	6
RC203	1049.62	3	1062.32	3	1050.64	3	941.81	5
RC204	798.46	3	799.08	3	798.46	3	801.87	四
RC205	1297.65	四	1303.68	四	1297.65	四	1165.82	七
RC206	1146.32	3	1155.33	3	1153.61	3	1072.85	5
RC207	1061.14	3	1095.37	3	1061.14	3	977.11	5
RC208	828.14	3	834.16	3	828.71	3	792.33	5

表 6
详细说明我们算法的结果，并与最知名的解决方案进行比较。

数据集	Best-known			This work						
	内华达州	推倒（网络用词）	作者	最佳道明	内华达州	平均时差	内华达州	缝隙	存储卡	中央处理器
C101	10	827.3	德罗谢尔、德罗谢尔和所罗门(1992)[12]	828.94	10	828.94	10	0	0	3592.54
C102	10	827.3	Desrochers 等人(1992 年)[12]	828.94	10	828.94	10	0	0	668.07
C103	10	826.3	塔瓦雷斯、马查多、佩雷拉和科斯塔(2003 年)[37]	828.07	10	828.07	10	0.00	0.00	252.88
C104	10	822.9	Tavares 等人(2003 年)[37]	824.78	10	827.93	10	0.54	5.57	140.18

C105	10	827.3	Tavares 等人(2003 年)[37]	828.94	10	828.94	10	0	0	1204.79
C106	10	827.3	Desrochers 等人(1992 年)[12]	828.94	10	828.94	10	0	0	1184.74
C107	10	827.3	Tavares 等人(2003 年)[37]	828.94	10	828.94	10	0	0	782.00
C108	10	827.3	Tavares 等人(2003 年)[37]	828.94	10	828.94	10	0	0	498.37
C109	10	827.3	Tavares 等人(2003 年)[37]	828.94	10	828.94	10	0	0	252.69
C201	3	589.1	库克和里奇(1999) [8]	591.56	3	591.56	3	0	0	4155.89
C202	3	589.1	库克和里奇(1999) [8]	591.56	3	591.56	3	0	0	630.48
C203	3	591.17	李和林(2003) [27]	591.17	3	591.31	3	0.35	3.66	278.99
C204	3	590.6	波特文和本吉奥(1996) [29]	594.89	3	603.16	3	2.95	10.12	152.45
C205	3	588.88	波特文和本吉奥(1996) [29]	588.88	3	588.88	3	0.00	0.00	1731.04
C206	3	588.49	Lau 等人(2003 年)[24]	588.49	3	588.49	3	0.00	0.00	1181.93
C207	3	588.29	罗沙特和泰勒德(1995) [30]	588.29	3	588.29	3	0	0	853.12
C208	3	588.03	谭, 周和李。(2006) [35]	588.32	3	588.32	3	0	0	476.75
R101	18	1607.7	Desrochers 等人(1992 年)[12]	1643.18	20	1645.82	20	0.39	5.48	1137.93
R102	17	1434	Desrochers 等人(1992 年)[12]	1460.26	18	1463.91	18.08	0.43	4.00	434.26
R103	13	1175.67	刘林刘(2001) [23] [21]	1217.39	15	1223.27	14.92	0.30	2.41	398.70
R104	10	974.2	谭等(2006) [35]	987.61	11	1002.54	11.33	0.92	6.04	227.74
R105	15	1346.12	Kallehauge, Larsen 和 Madsen (2006) [19]	1363.91	15	1372.01	15.75	0.68	5.45	641.26
R106	13	1234.6	库克和里奇(1999) [8]	1247.90	13	1256.45	13.42	0.93	4.76	457.90
R107	11	1051.84	Kallehauge 等人(2006 年)[19]	1087.50	12	1097.41	12.00	1.03	4.46	336.94
R108	10	954.03	谭等(2006) [35]	961.85	11	965.82	10.83	1.25	5.91	220.71
R109	12	1013.2	蒋和罗素(1997) [7]	1152.99	13	1163.07	13.00	1.23	7.20	365.31
R110	12	1068	库克和里奇(1999) [8]	1091.50	12	1100.83	12.17	1.49	8.51	340.76
R111	12	1048.7	库克和里奇(1999) [8]	1067.46	12	1076.61	12	0.98	7.16	302.90
R112	10	953.63	罗沙特和泰勒德(1995) [30]	973.25	10	978.16	10.67	0.71	6.47	240.36
R201	8	1198.15	谭等(2001) [36]	1174.69	6	1178.90	6.08	0.46	4.85	547.67
R202	6	1077.66	谭等(2001) [36]	1046.10	5	1053.44	5.08	1.41	7.88	331.63
R203	5	933.286	谭等(2001) [36]	884.02	5	896.05	4.92	1.43	7.77	273.62
R204	3	752.13	Barbucha (2014 年)[2]	750.40	四	758.13	四	1.52	6.50	230.77
R205	3	994.42	卢梭、金德罗和佩桑特(2002) [31]	960.75	5	975.83	4.92	1.36	7.61	290.46
R206	3	833	坦吉亚、奥斯曼和孙(1994) [38]	900.97	四	908.18	4.25	1.61	6.76	226.36
R207	3	814.78	罗沙特和泰勒德(1995) [30]	809.72	四	826.74	四	2.15	11.31	234.11
R208	2	729.42	Barbucha (2014 年)[2]	723.14	5	732.31	3.42	1.37	6.99	198.33
R209	3	855	Thangiah 等人(1994 年)[38]	863.12	5	882.22	4.92	1.35	6.71	226.84
R210	3	943.10	Barbucha (2014 年)[2]	927.54	5	938.63	4.92	2.09	9.31	232.11
R211	2	767.82	Barbucha (2014 年)[2]	763.22	四	781.23	四	2.18	7.61	189.63
RC101	15	1619.8	科尔、德罗西亚斯、麦德森、所罗门和索米斯(1999) [22]	1646.17	16	1656.01	16.33	0.66	6.93	663.29
RC102	13	1470.26	谭等(2006) [35]	1481.61	14	1488.79	14.83	0.81	7.29	441.74
RC103	12	1196.12	谭等(2006) [35]	1280.76	12	1298.32	12.08	1.94	18.50	308.63
RC104	10	1135.48	科尔多、拉波特和梅塞尔(2001) [9]	1162.03	11	1168.13	11.00	0.60	5.43	217.15
RC105	14	1542.29	Barbucha (2014 年)[2]	1545.30	16	1554.79	16.17	1.57	8.50	545.30
RC106	13	1371.69	谭等(2006) [35]	1401.17	14	1413.38	14.00	0.82	7.41	301.56
RC107	11	1222.16	谭等(2006) [35]	1235.28	12	1256.98	12.50	1.96	11.91	277.52
RC108	11	1133.82	罗、李、陈、刘(2015) [28]	1136.35	11	1149.65	11.17	1.67	9.76	235.20
RC201	6	1134.91	谭等(2006) [35]	1271.78	七	1284.59	6.92	0.78	5.71	476.51
RC202	5	1130.53	谭等(2006) [35]	1116.21	6	1122.97	5.75	0.71	5.25	350.15
RC203	四	1026.61	谭等(2006) [35]	941.81	5	951.30	5.08	0.79	6.33	226.52
RC204	3	799.08	Barbucha (2014 年)[2]	801.87	四	809.09	四	2.22	8.13	162.89
RC205	5	1295.46	谭等(2006) [35]	1165.82	七	1172.80	七	2.23	16.34	335.38
RC206	四	1112.2	于、杨和姚。(2011) [42]	1072.85	5	1082.93	5.50	1.15	8.40	272.82
RC207	四	1040.67	谭等(2006) [35]	977.11	5	998.46	5.42	2.34	9.82	211.82
RC208	3	829.69	Rousseau 等人(2002) [31]	792.33	5	809.23	4.67	2.11	8.65	191.15

间隙=(平均 TD-最佳 TD)/最佳 TD, 标清为 15 倍 TD 的标准偏差。

Table 7
Detail comparison between two parameters setting.

Data set	Tabu tenure = 30, Tabu iteration times = 500, ABC iteration times = 200n, similarity restriction = 0.7 and update restriction = 10n					Tabu tenure = 30, Tabu iteration times = 500, ABC iteration times = 200n, similarity restriction = 0.6, and update restriction = 50n				
	Best TD	NV	Mean TD	CPU		Best TD	NV	Mean TD	NV	CPU
C101	828.94	10	828.94	3592.54		828.94	10	828.94	10	1070.42
C102	828.94	10	828.94	668.07		828.94	10	828.94	10	280.49
C103	828.07	10	828.07	252.88		828.07	10	828.07	10	116.66
C104	824.78	10	827.93	140.18		824.78	10	829.26	10	60.94
C105	828.94	10	828.94	1204.79		828.94	10	828.94	10	478.01
C106	828.94	10	828.94	1184.74		828.94	10	828.94	10	389.41
C107	828.94	10	828.94	782.00		828.94	10	828.94	10	322.39
C108	828.94	10	828.94	498.37		828.94	10	828.94	10	185.52
C109	828.94	10	828.94	252.69		828.94	10	828.94	10	83.06
C201	591.56	3	591.56	4155.89		591.56	3	591.56	3	1337.71
C202	591.56	3	591.56	630.48		591.56	3	591.56	3	348.48
C203	591.17	3	591.17	278.99		593.21	3	593.21	3	132.09
C204	594.89	3	603.16	152.45		590.6	3	608.03	3	53.45
C205	588.88	3	588.88	1731.04		588.88	3	588.88	3	619.37
C206	588.49	3	588.49	1181.93		588.49	3	588.49	3	473.96
C207	588.29	3	588.29	853.12		588.29	3	588.29	3	322.58
C208	588.32	3	588.32	476.75		588.32	3	588.32	3	234.39
R101	1643.18	20	1645.82	1137.93		1644.5	20	1650.89	20.27	521.88
R102	1460.26	18	1463.91	434.26		1463.52	18	1469.81	18.07	187.46
R103	1217.39	15	1223.27	398.70		1224.44	15	1228.17	14.8	141.13
R104	987.61	11	1002.54	227.74		1001.89	12	1011.12	11.8	85.29
R105	1363.91	15	1372.01	641.26		1369.07	16	1378.44	15.93	235.14
R106	1247.90	13	1256.45	457.90		1251.71	13	1263.34	13.73	174.85
R107	1087.50	12	1097.41	336.94		1093.99	12	1105.3	11.93	130.82
R108	961.85	11	965.82	220.71		958.44	11	970.46	10.87	87.48
R109	1152.99	13	1163.07	365.31		1160.06	13	1174.37	13.27	162.67
R110	1091.50	12	1100.83	340.76		1095.34	12	1111.61	12.13	154.76
R111	1067.46	12	1076.61	302.90		1073.29	12	1083.78	12.2	131.07
R112	973.25	10	978.16	240.36		981.67	11	988.63	10.87	118.97
R201	1174.69	6	1178.90	547.67		1178.91	6	1184.33	6.07	196.05
R202	1046.10	5	1053.44	331.63		1041.48	5	1056.15	5.2	119.36
R203	884.02	5	896.05	273.62		889.4	5	902.09	4.53	85.53
R204	750.40	4	758.13	230.77		748.15	4	759.52	4	69.77
R205	960.75	5	975.83	290.46		968.89	5	982.09	4.8	116.61
R206	900.97	4	908.18	226.36		897.2	4	911.63	4	78.55
R207	809.72	4	826.74	234.11		812.61	4	830.07	4	73.13
R208	723.14	5	732.31	198.33		725.58	4	735.53	3.13	65.18
R209	863.12	5	882.22	226.84		872.65	5	884.44	4.93	90.78
R210	927.54	5	938.63	232.11		919.41	5	938.66	4.8	95.88
R211	763.22	4	781.23	189.63		769.28	4	786.04	4	77.49
R212	1646.17	16	1656.01	663.29		1650.2	16	1661.04	16.4	271.72
RC101	1481.61	14	1488.79	441.74		1481.48	14	1493.5	14.8	193.72
RC102	1280.76	12	1298.32	308.63		1288.36	12	1313.42	12.13	143.69
RC103	1162.03	11	1168.13	217.15		1174.46	11	1181.55	11.27	92.34
RC104	1545.30	16	1554.79	545.30		1535.94	15	1560.12	16.53	195.66
RC105	1401.17	14	1413.38	301.56		1411.51	14	1423.01	13.87	163.77
RC106	1235.28	12	1256.98	277.52		1236.27	12	1260.54	12.07	137.43
RC107	1136.35	11	1149.65	235.20		1131.4	11	1150.3	11.4	132.3
RC108	1271.78	7	1284.59	476.51		1285.14	7	1295.15	6.73	174.15
RC201	1116.21	6	1122.97	350.15		1116.58	6	1124.5	5.87	122.95
RC202	941.81	5	951.30	226.52		944.87	5	952.79	5	82.79
RC203	801.87	4	809.09	162.89		791.76	4	809.33	4.2	56.04
RC204	1165.82	7	1172.80	335.38		1163.29	7	1189.25	6.6	115.08
RC205	1072.85	5	1082.93	272.82		1070.82	6	1083.09	5.47	98.9
RC206	977.11	5	998.46	211.82		980.87	5	1003.8	5.13	85.6
RC207	792.33	5	809.23	191.15		799.11	4	816	4.73	81.55

参考

- [1] 南. 容量受限车辆路径问题的混合大邻域搜索算法, 专家系统. 应用. 61 (2016) 28–38.
- [2] D. 带时间窗的车辆路径问题的合作种群学习算法, 神经计算 146(2014)210–229.
- [3] 通用软时间窗车辆路径问题的一种新的混合列生成元启发式方法. Sci. 316 (20) (2015) 598–615.
- [4] A. 博尔特费尔特, t. 哈恩, d. 梅内尔, L.M. Hybrid, 具有集群回程和三维装载约束的车辆路径问题的算法, 欧洲. J. 奥佩. Res. 243 (1) (2015) 82–96.
- [5] J. 卡塞雷斯-克鲁兹, p. 阿里亚斯, d. 吉马兰, d. 瑞拉, 人工智能. Juan, Rich 车辆路径问题: 调查, ACM Comput. 生存. 47 (2) (2015) 32:1–28.
- [6] 米 (meter 的缩写). 奇克斯利, g. s. 德萨尼尔. Irnich, g. 有时间窗和多栈的提货和交货问题的分支降价算法. J. 奥佩. Res. 250 (3) (2016) 782–793.
- [7] 求解带时间窗车辆路径问题的反应式禁忌搜索元启发式算法. 电脑. 9 (4) (1997) 417–430.
- [8] W. 带时间窗的车辆路径问题的平行切割平面算法, 计算和应用数学系, 莱斯大学, 休斯敦, 德克萨斯州, 1999 年技术报告.
- [9] J. -F. Cordeau. 拉波特, a. 有时间窗的车辆路径问题的统一禁忌搜索启发式算法. 奥佩. Res. 社会主义者. 52 (8) (2001) 928–936.
- [10] J. -F. Cordeau. 车辆路径问题的并行迭代禁忌搜索启发式算法. 奥佩. Res. 39 (2012) 2033–2050. [11] 丹齐格, 拉姆塞, 卡车调度问题, 管理. Sci. 6 (1) (1959) 80–91.
- [12] 米 (meter 的缩写). j. 德罗斯彻斯. 德罗西亚斯先生. 有时间窗的车辆路径问题的一种新的优化算法. Res. 40 (2) (1992) 342–354.
- [13] 米 (meter 的缩写). Gendreau. 赫兹, g. 车辆路径问题的禁忌搜索启发式算法. Sci. 40 (10) (1994) 1276–1290.
- [14] 米 (meter 的缩写). Gendreau. 八神庵. 拉波特, s. 一个路由和集装箱装载问题的禁忌搜索算法, 运输. Sci. 40 (3) (2006) 342–350.
- [15] 龚永杰, 美国张军. 刘. -Z. 黄. 纽约钟. -H. 史, 带时间窗的车辆路径优化问题: 离散粒子群优化方法, IEEE Trans. 系统. 曼赛博. 第 c42(2)(2012)254–267 部分.
- [16] 米 (meter 的缩写). Hifi, 我. Kacem, s. 内格雷, l. 吴, 三维装箱问题的线性规划方法. 注释 Discr. 数学. 36 (2010) 993–1000.
- [17] 米 (meter 的缩写). Hifi, l. 吴, 带时间窗的车辆路径问题的混合元启发式算法, 载于: 2014 国际控制、决策和信息技术会议 (CoDIT), Metz, 2014, pp. 188–194.
- [18] 《自然和人工系统中的适应性: 生物学、控制和人工智能应用的介绍性分析》, 麻省理工学院出版社, 剑桥, 1992 年.
- [19] B. Kallehauge. 拉格朗日对偶在有时间窗的车辆路径上的应用. 奥佩. Res. 33 (5) (2006) 1464–1487.
- [20] D. 卡拉博加, 基于蜜蜂群的数值优化思想, 技术报告 TR06, 埃尔基耶斯大学, 2005.
- [21] 南. 模拟退火优化: 定量研究. 统计一下. 物理. 34 (1984) 975–986.
- [22] 名词 (noun 的缩写). j. 科尔. m. o. b. g. 马德森, Desrosiers. 所罗门, f. 有时间窗的车辆路径问题的双向切割, 运输. Sci. 33 (1) (1999) 101–116.
- [23] 刘汉忠, 林永福, 刘庆忠, 基于约束的局部搜索的搜索领域多样化及其在 VRPTW 的应用, 第三届人工智能与人工智能技术集成国际研讨会, 2001.
- [24] 刘汉忠先生. 有时间窗和有限数量车辆的车辆路径问题, 欧元. J. 奥佩. Res. 148 (3) (2003) 559–569.
- [25] 梁世华先生. 张, d. 张, x. 二维装载约束下异构车队车辆路径问题的元启发式算法. J. 奥佩. Res. 225 (2) (2013) 199–210.
- [26] 梁世华, x. 周, 马超. 张, j. 郑, 二维装载车辆路径问题的扩展引导禁忌搜索和一种新的装箱算法, Comput. 奥佩. Res. 38 (1) (2011) 205–215.
- [27] H. 李, a. Lim, 类似退火的局部搜索重新开始求解 VRPTW, Eur. J. 奥佩. Res. 150 (1) (2003) 115–127.
- [28] 罗, x. 李, 陈敏瑞, 刘华伟, 一种求解带时间窗车辆路径问题的混合混合蛙跳算法, Inf. Sci. 316 (2015) 266–292.
- [29] J.Y. Potvin, s. 有时间窗的车辆路径问题第二部分: 遗传搜索. 电脑. 8 (2) (1996) 165–172.
- [30] Y. 车辆路线局部搜索中的概率多样化和强化. 赫里斯特. 1 (1) (1995) 147–167.
- [31] 卢梭先生. Gendreau. 用基于约束的算子解决带时间窗的车辆路径问题. 赫里斯特. 8 (1) (2002) 43–58.
- [32] 有时间窗约束的车辆路径和调度问题的算法. Res. 35 (2) (1987) 254–265.
- [33] 司徒伟, 吴永忠, SinC. 求解容量受限车辆路径问题的人工蜂群算法. J. 奥佩. Res. 215 (1) (2011) 126–135.
- [34] É. 尾灯, p. Badeau. Gendreau. 软时间窗车辆路径问题的禁忌搜索启发式算法, 运输. Sci. 31 (2) (1997) 170–186.
- [35] 求解带时间窗车辆路径问题的混合多目标进化算法. Optim. 应用. 34 (1) (2006) 115–151.
- [36] 谭国成, 李立宏, 金. 欧, 人工智能启发式算法在求解带时间窗约束的车辆路径问题中的应用. 应用. Artif. 智力. 14 (6) (2001) 825–837.
- [37] J. 塔瓦雷斯, p. 马查多, 佩雷拉, 科斯塔, 论 GVR 对车辆路线的影响, 载于: 2003 年美国计算机学会应用计算研讨会论文集, 纽约, 2003 年, 页. 753–758.
- [38] S.R. Thangiah, I.H. Osman 和 t. 孙, 混合遗传算法, 带时间窗车辆路径问题的模拟退火和禁忌搜索方法, 计算机科学系, 滑岩大学, 技术报告 CpSc-TR-94-27, 1994.
- [39] 长度. 魏, z. 张, l. 具有三维装载约束的异构车队车辆路径问题的自适应可变邻域搜索. 智力. 玛格. 9 (4) (2014) 18–30.
- [40] 姚. 胡, 张敏洪, s. 王, 带扫描策略的人工蜂群算法求解周期车辆路径问题, 仿真 89(6)(2013)762–770.
- [41] m. e. t. Yassen. 带时间窗车辆路径问题的元和谐搜索算法. Sci. 325 (2015) 140–158.
- [42] B. 俞, 杨振中, 姚宝中, 带时间窗的车辆路径问题的混合算法, 专家系统. 应用. 38 (1) (2011) 435–441.
- [43] E.E. Zachariadis, C.D. Tarantilis, C.T. Kiranoudis, 在时间窗和装载限制下为不同请求类型的混合设计车辆路线, Eur. J. 奥佩. Res. 229 (2) (2013) 303–317.
- [44] D. 张, l. 魏, 问. 陈 h. 陈, 三维装箱问题的组合启发式算法. 软. 18 (9) (2007) 2083–2089.
- [45] Z. 张, l. 魏, a. 三维装载约束下最小化燃料消耗的能力受限车辆路径问题的进化局部搜索, 运输. Res. b82(2015)20–35 部分.