

Research on Automobile Parts Milk-Run Routing with Unload Time Window

Yehui Tao^{1,2}, Lei Zhao^{1,2}, Daoli Zhu^{1,2}

¹Sino-US Global Logistics Institute, Shanghai Jiaotong University, Shanghai

²Antai College of Economics & Management, Shanghai Jiaotong University, Shanghai

Email: teresi93@outlook.com

Received: Dec. 8th, 2016; accepted: Dec. 22nd, 2016; published: Dec. 29th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

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

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



Open Access

Abstract

Nowadays, pulling type production becomes more popular in automobile supply chain. Inbound logistics of auto parts showed a high degree of complexity and professionalism. For the milk-run problem in such scene, this paper designs a model with unload time window and split deliveries, to minimize the transportation cost. According to the characteristics of the model, this paper designed a tabu search algorithm, and the initial solution and the neighborhood search algorithm are improved. Finally, it proved the validity of the algorithm by the experimental data, and compared with the existing algorithms. The result shows that the milk-run mode can effectively save transportation cost and ensure production efficiency.

Keywords

Automobile Parts Supply Logistics, Milk-Run, Unload Time Windows, Tabu Research Algorithm

带卸货时间窗的循环取货路径计划问题研究

陶业辉^{1,2}, 赵磊^{1,2}, 朱道立^{1,2}

¹上海交通大学中美物流研究院, 上海

²上海交通大学安泰经济与管理学院, 上海

Email: teresi93@outlook.com

收稿日期: 2016年12月8日; 录用日期: 2016年12月22日; 发布日期: 2016年12月29日

摘要

当今汽车供应链拉动式生产盛行，汽车零部件入厂物流表现出高度的复杂性和专业性，针对该场景下的循环取货问题，引入卸货时间窗的概念，设计供应商集货需求可拆分的数学模型，以降低零部件循环取货过程中的车辆运输成本，同时满足卸货时间窗和车辆容量的限制。根据所设计模型的特点设计禁忌搜索算法，改进初始解及邻域搜索算法，最后通过实验数据证明算法的有效性，并与现有算法进行比对，表明该模式下的循环取货模式可以更有效的节约运输成本，保证生产效率。

关键词

汽车零部件入厂物流，循环取货，卸货时间窗，禁忌搜索算法

1. 引言

汽车零部件入厂物流具有高度复杂性和专业性，优化的汽车零部件入厂物流可以降低物流成本且同时提高响应市场的速度和生产柔性，因此它是汽车制造企业增强供应链竞争能力的重要环节。在汽车供应链拉动式结构、零部件采购全球化、整车生产准时化的背景下汽车生产物料供给逐渐形成了递阶供应的模式。汽车零部件一般经过三/四阶递阶供应从供应商到达生产线终端，如图 1 所示。第一阶集货物流，为零部件从海外供应商经 Milk-run 循环取料运至海外的集货中心；第二阶取货物流，为零部件从海外集货中心经干线运输运至国内配送中心，或者为零部件从国内供应商经 Milk-run 循环取料运至国内的配送中心；第三阶配送物流，为零部件从国内配送中心经配送运输运至国内主机厂货架；第四阶场内物流，为零部件从主机厂货架经厂内运输运至国内汽车装配线。在这个过程中，第一阶集货物流与第二阶取货物流中都存在着 Milk-run 循环取料问题，因而 Milk-run 是汽车零部件入厂物流合理化的重要途径之一，它不仅能够解决由供应商单独配送带来的时间和运输成本问题，并且还能够大幅度提高汽车零部件入厂物流的效率。

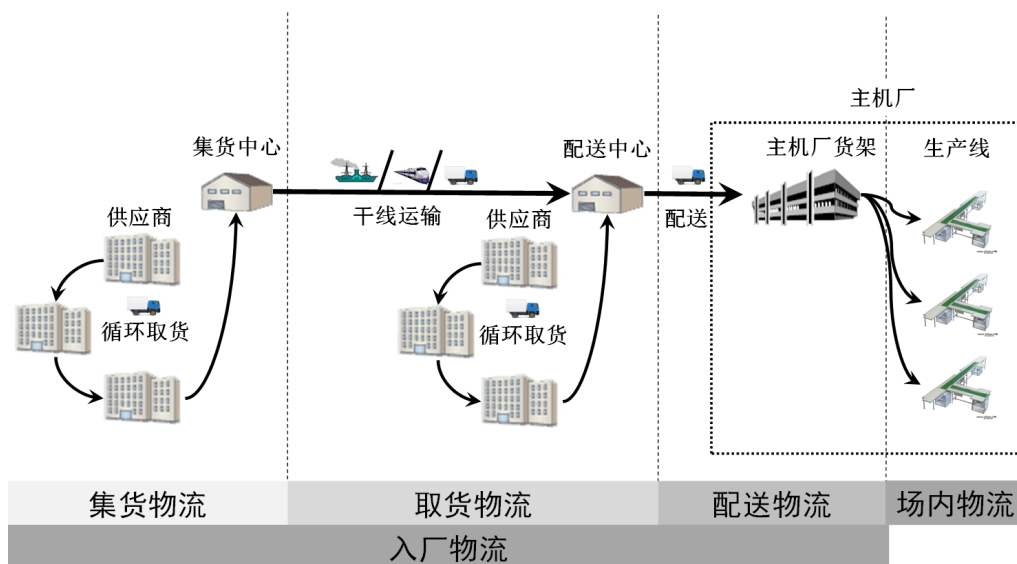


Figure 1. The automobile parts inbound supply chain
图 1. 汽车零部件递阶供应示意图

Milk-Run 的概念源自奶制品行业, 又称循环取货、牛奶式取货等, 它的运作方式是车辆从配送中心出发, 按照既定的路线和时间要求依次到不同的供应商处收取原料或者零配件, 并将其集中到配送中心然后再进行拼装配送到主机厂。Milk-Run 的显著优点是能够满足小批量多频次的取货需求并且能够提高车辆的装载率和使用效率, 有效减少装配工厂的拥挤、大幅度减少车辆总行驶距离、汽车尾气的排放等。Browsersox 等人[1]指出 milk-run 取货模式是精益物流战略的重要组成部分。Du 等人[2]研究了一个集货 milk-run 取货中心的实时车辆调度系统, 并对初始参数进行了分析。文献[3] [4] [5]指出 milk-run 取货模式能够大幅度减少运输距离和物流成本, 提高车辆运输效率与减少二氧化碳排放, 减少车辆使用量进而改善城市交通状况。但在他们的研究中没有进行针对汽车零部件入厂中 milk-run 取货模式特征的分析。

汽车零部件中循环取货模式应用的关键是在相关约束条件下的取货路径优化。在汽车零部件入厂物流的实际运作中, 处于供应链的核心地位的汽车主机厂奉行 JIS、JIT 等拉动式生产方式, 力求降低自身库存水平, 因此对所需零部件到达主机厂的时间做严格的要求。与此同时, 为了提升运输环节的车辆利用率, 一个供应商的货物可以由多个批次的车辆进行载运, 这些情景与传统循环取货模式有一定区别。因此本文针对循环取货在汽车零部件入厂物流中的应用, 提出一种新的适应汽车零部件入厂物流中 Milk-run 循环取料问题特点的混合整数规划模型, 并对模型进行分析。同时, 本文提出一种启发式算法对该问题进行求解, 并通过算法分析和数值实验论证了本文提出的方法可以有效地解决该问题。本文结构如下: 第二节对 Milk-run 循环取料问题与传统车辆路径问题(VRP 问题)进行比较分析, 提出 Milk-run 循环取料问题的新特点。第三节本文针对 Milk-run 循环取料问题构建了混合整数规划模型。第四节针对 Milk-run 循环取料问题提出一种启发式方法进行求解。第五节通过计算实验分析论证了本文方法的有效性。第六节对全文进行总结。

2. 问题描述

车辆路径问题最早在 1959 年由 Dantzig 和 Ramser 提出, 用来描述对天然气服务站的配送。1964 年, Clark 和 Wright 提出了一种有效的贪婪算法, 提升了 Dantzig 和 Ramser 对该问题的求解效率, 在这之后的 50 多年中, 大量的研究工作对车辆路径问题进行了研究。汽车零部件入厂 Milk-run 循环取料问题与车辆路径问题有很大的相似之处。本节将对汽车零部件入厂 Milk-run 循环取料问题与车辆路径问题之间的异同进行比较分析。

汽车零部件入厂 Milk-run 循环取料问题是在满足约束条件的情况下找到总运输成本尽量少的路线方案, 其应当满足的约束条件如下: (1) 所有车辆都从配送中心出发最终回到配送中心; (2) 每一个供应商的需求取货量可以被一辆或者多辆车取回; (3) 每一个供应商的需求取货量都存在一个卸货要求时间窗; (4) 每一辆车的总取货量都不能超过其对应车型的载货能力。

与汽车零部件入厂 Milk-run 循环取料问题类似的车辆路径问题主要有: 带时间窗的车辆路径问题、需求可拆分的车辆路径问题、多车型车辆路径问题。带时间窗的车辆路径问题由 Savelsbergh [6]最早证明是一个 NP-complete 问题。之后的研究中, Golden [7], Solomon [8]主要集中于对 VRPTW 问题可应用求解方法的研究, 近年, Raul Banos 等人[9]提出了一种针对多目标 VRPTW 问题的混合现代启发式算法, 得到优化解决方案。但在这些研究中, 并未考虑到卸货时间窗和需求可拆分的特点。需求可拆分的车辆路径问题由 Dror 和 Trudeau [10]提出, 之后 Archetti 等人[11]对应用规模的这一问题进行了启发式方法的研究, 国内方面, 孟凡超[12]、熊浩[13]建立了需求可拆分车辆路径问题的数学模型, 并设计禁忌搜索算法求解, 但这些研究中并未考虑到卸货时间窗的特点。因此汽车零部件入厂 Milk-run 循环取料问题可以看作是一种新的车辆路径问题的推广, 汽车零部件入厂 Milk-run 循环取料问题是一个 NP-hard 问题。

3. 问题定义与模型

3.1. 定义

3.1.1. 集合

本文的研究的汽车零部件入厂物流循环取货问题可以被定义在一个全连通图 $G=(V,E)$ 中, 集合定义如下:

$V = \{0, \dots, n+1\}$	节点集合, 其中 0 和 $n+1$ 为配送中心
A	弧集合
$N = V \setminus \{0, n+1\}$	供应商集合
$\Delta^+(i)$	节点 i 的前向路径集合
$\Delta^-(i)$	节点 i 的后向路径集合
$K = \{1, \dots, m\}$	车型集合, 容量按从大到小进行排列
$P = \{1, \dots, l\}$	每种车型的车辆编号集合

3.1.2. 参数

在以上集合定义的基础上, 本模型相关参数定义如下:

C_{ij}	节点 i 与节点 j 之间的运输成本
T_{ij}	节点 i 与节点 j 之间的运输时间
D_i	供应商 $i \in V$ 的取零部件量, 其中 $D_0 = D_{n+1} = 0$
S_i	每个供应商的装货时间 $i(i=1, \dots, n)$, 其中 $S_0 = S_{n+1} = 0$
Cap	车辆的载货能力
E_i	供应商 $i \in V$ 的提供零部件的道口最早卸货时间, 其中 $E_0 = E_{n+1} = 0$
L_i	供应商 $i \in V$ 的提供零部件的道口最晚卸货时间, 其中 $L_0 = L_{n+1} = \max_{i \in V} (L_i)$
M	一个足够大的数

3.2.3. 变量

基于以上定义, 本模型引入三个变量, 分别用来对路径、访问时间和去货量进行决策:

x_{jk}	如果弧 $(i, j) \in A$ 被选为编号为 $k \in K$ 的车辆运行, 则为 1, 否则为 0
w_{ik}	节点 $i \in V$ 被编号为 $k \in K$ 的车辆的开始服务时间
v_{ik}	编号为 $k, k \in K$ 的车辆从供应商 $i, i \in V$ 对零件的取货量

3.2. 模型

本问题所构建目标模型的目标函数及约束条件可以表述如下:

$$\begin{aligned}
\text{Min} \quad & f(x) = \sum_{k \in K} \sum_{(i,j) \in A} C_{ij} x_{ijk} & (1) \\
\text{s.t.} \quad & \sum_{k \in K} \sum_{(i,j) \in A} C_{ij} x_{ijk} \geq 1, \forall i \in N & (2) \\
& \sum_{j \in \Delta^+(0)} x_{0jk} = 1, \forall k \in K & (3) \\
& \sum_{i \in \Delta^-(j)} x_{ijk} - \sum_{i \in \Delta^+(j)} x_{jik} = 0, \forall k \in K, j \in N & (4) \\
& \sum_{i \in \Delta^-(n+1)} x_{i,n+1,k} = 1, \forall k \in K & (5) \\
& w_{ik} + S_i + T_{ij} - w_{jk} \leq (1 - x_{ijk})M, \forall k \in K, (i, j) \in A & (6) \\
& E_i y_{ik} \leq w_{n+1,k} \leq L_i y_{ik} + M(1 - y_{ik}), \forall k \in K, i \in N, & (7) \\
& v_{ik} \leq D_i \sum_{j \in V} x_{ijk}, \forall i \in N, k \in K & (8) \\
& \sum_{k \in K} v_{ik} = D_i, \forall i \in N & (9) \\
& \sum_{i \in V} v_{ik} \leq \text{Cap}, \forall k \in K & (10) \\
& x_{ijk} \in \{0, 1\}, \forall k \in K, (i, j) \in A & (11) \\
& w_{ik} \geq 0, v_{ik} \geq 0, \forall k \in K; i \in V & (12)
\end{aligned}$$

3.3. 模型描述

式(1)为目标函数，表示总成本，即总运输里程成本。式(2)~式(12)为约束条件，其中式(2)表示每个供应商至少被一辆车访问一次；式(3)、式(5)表示所有车辆都由 RDC 出发，最终回到 RDC；式(4)表示访问某个供应商的车辆都一定会离开该供应商；式(6)表示车辆运行的时间顺序，用来避免闭回路的出现；式(7)表示访问供应商的车辆需要满足该供应商的卸货时间窗；式(8)、式(9)表示任意供应商的供货量只能被拆分到经过该供应商的车辆上，且该供应商的供应量需要被完全运载；式(10)表示车辆的总装载量应满足该线路对应车辆的装载能力约束。

4. 算法设计

4.1. 算法框架

上述模型基于现实汽车零部件入厂物流的情景，在传统车辆路径问题(vehicle routing problem)基础上进行延伸，引入需求可拆分以及卸货时间窗等要素，属于 NP-Hard 问题，难以在合适的时间内得到精确解，因此本文设计改进的启发式算法进行求解，初始解的构造将采用改进的最佳插入法，并运用禁忌搜索算法进一步优化。

4.2. 初始解生成

禁忌搜索算法多初始解依赖性较强，本文将本文采用改进的最优插入算法进行初始解的生成。具体流程如下：

Step 1: 集合 U 为未访问点的集合，判断 U 是否为空集，若是则终止；否则转 Step 2。

Step 2: U 中未访问点按照其距离 RDC 的距离从大到小进行排序，排序列表为 S 。

Step 3: 选择 S 排序中第一个供应商节点来建立一条由 RDC 出发经过该供应商节点并返回 RDC 的初始路径。

Step 4: 考察此时路径的总容量是否满载，否则将此供应商阶段从集合 U 以及列表 S 中删除，并进行 Step 6，是则转入 Step 7。

Step 5: 考察此时路径是否超载, 是否超载, 否则该路径构建结束, 将最后插入路径的供应商从集合 U 以及列表 S 中删除, 是则将店进行拆分, 满足路径满载, 构建结束, 超出部分保留在 U 中, 转 Step 1。

Step 6: 对 U 中任意节点 u 分别考察其插入现有路径中任意两点 (i, j) 之间时是否满足时间窗约束, 将满足条件的节点及其对应位置设为集合。

Step 7: 对 U 中任一节点 u 分别计算将其插入现有路径中任意时间窗可行位置 (i, j) 之间造成的成本增加值 $c(i, u, j) = d_{iu} + d_{uj} - d_{ij}$, 选择具有最小增加值的 $c(i_u, u, j_u)$ 的位置 (i_u, j_u) 作为该节点的最佳插入位置, 对 U' 中所有点的最佳插入位置进行比并选取出具有最小成本增加值的节点 u^* , 将其插入到最佳位置即路径 (i_{u^*}, j_{u^*}) , 转 Step 4。

4.3. 禁忌搜索算法设计

禁忌搜索算法最早是由美国的 Glover 提出, 它是对局部邻域搜索的一种改进, 引入禁忌表, 通过该方法来避免因重复搜索而陷入局部最优。每当达到一个局部最优点, 就利用禁忌表将该点记录下来, 在下次邻域搜索中, 可以有意识的回避这些禁忌表中的局部最优点。因其具有这种禁忌特性, 所以相较于局部邻域搜索算法, 禁忌搜索算法更易于寻得全局最优。

在禁忌搜索算法的设计过程中, 主要的环节包括以下几点: 解的表示、解的评价、邻域的构造、禁忌对象及禁忌长度的确定、特赦规则的确定和终止规则的确定。

4.3.1. 解的表示

本实验的解采用需求带与需求量对应分别排列的方式进行表示, 用代表 RDC, 代表各个供应商节点, 对应路线用同样顺序构建通过各供应商时的取货量。例如, 路径 0-3-4-1-0 及其对应装载 0-1-1-2-0 表明该路线途经 3、4、1 供应商节点, 并依次获取 1、1、2 单位货物。

4.3.2. 解的评价

本实验以目标函数作为解的评价函数, 目标函数的值越小, 代表解得质量越高。

4.3.3. 邻域构造

VRP 问题的邻域搜索主要有两种, 一个是线路内的搜索, 另一个是线路间的搜索。本文采用 or-opt 交换法以及 λ -interchange 法进行邻域搜索, 具体方法如下:

1) or-opt 交换法用于线路内的搜索。即在一条路径上选取一点 i , 寻找其在该路线的最优位置并进行插入操作, 且要满足时间窗等约束。由于该操作仅仅是改变了该路径上各点的访问顺序, 而总体供货需求量没有改变, 因此对该路径而言, 容量限制总是可以被保证的, 也就不会出现新的拆分或者合并拆分的现象。

2) λ -interchange 法用于线路间的搜索, 具体又可分为 1-0 节点交换法、1-1 节点交换法和 1-2 节点交换法。1-0 节点交换法是指将某一供应商点从其当前所在的路线中删除, 然后将其插入到另一条路径中; 1-1 节点交换法是指从两条路径中分别选择两点, 然后互换它们的位置; 1-2 节点交换法是指从一条路径中选择一点, 再从另一路径中选择两点, 互换这一点与这两点的位置。本文在研究过程中将采用 λ -interchange 的 1-0 节点交换法和 1-1 节点交换法, 并根据本研究需求可拆分及卸货时间窗约束的特点, 对具体操作进行一定的改进。

1-0 节点交换法即将一条路径上的某一点 i 首先从路径 r_1 上删除, 然后再将其插入到另一路径 r_2 的最优位置, 同时要满足 i 在路径 r_2 上的卸货时间窗及容量等约束。针对本研究特点, 具体可分为以下两种情

况讨论。

Senior1: $i \in r_1, i \notin r_2$, 计算 r_2 路径的剩余容量 Q^2 , 若 $Q^2 \geq D_i^1$, 则将 i 从路径 r_1 中删除, 插入到路径 r_2 中去, 并更新 $D_i^2 = D_i^1, D_i^1 = 0$; 若 $Q^2 < D_i^1$, 则将 i 插入到路径 r_2 中, 同时将其需求拆分为 $D_i^2 = Q^2, D_i^1 = D_i^1 - Q^2$ 。

Senior2: $i \in r_1, i \in r_2$, 计算 r_2 路径的剩余容量 Q^2 , 若 $Q^2 \geq D_i^1$, 则将 i 从路径 r_1 中删除, 并更新 $D_i^2 = D_i^1 + D_i^2, D_i^1 = 0$; 若 $Q^2 < D_i^1$, 此时无法进行插入操作。

1-1 交换法即将一条路径 r_1 上的某一点 i 与另一路径 r_2 上的某点 j 的位置进行交换, 同时需满足各项约束条件。针对所研究问题, 具体可分为以下三种情况:

Senior3: $i \in r_1, i \notin r_2, j \in r_2, j \notin r_1$, 计算 r_2 路径的剩余容量 Q^2 , 若 $Q^1 + D_i^1 \geq D_j^2, Q^2 + D_j^2 \geq D_i^1$, 则将 i, j 两点完全互换, 并更新 $D_i^2 = D_i^1, D_j^1 = D_j^2$, 若 $Q^1 + D_i^1 < D_j^2, Q^2 + D_i^2 > D_j^1$, 则将 i 从路径 r_1 中删除, 插入到路径 r_2 中, $D_i^2 = D_i^1$, 将 j 插入到 r_1 中, 同时将其需求拆分为 $D_j^1 = D_i^1 - Q^2, D_j^2 = D_j^2 - D_i^1 + Q^2$ 。

Senior4: $i \in r_1 \cap r_2, j \in r_2, j \notin r_1$, 且 $D_i^1 \leq D_j^2$, 则将 i 从路径 r_1 中删除, 并更新 $D_i^1 = 0, D_i^2 = D_i^1 + D_i^2$, 同时将 j 插入到 r_1 中, 同时将其需求拆分为 $D_j^1 = D_i^1, D_j^2 = D_j^2 - D_i^1$ 。

Senior5: $i \in r_1 \cap r_2, j \in r_2 \cap r_1$, 且 $D_i^1 \leq D_j^2$, 则将 i 从路径 r_1 中删除, 并更新 $D_i^2 = D_i^1 + D_i^2$, j 点位置不变, 同时将其需求拆分为 $D_j^1 = D_i^1, D_j^2 = D_j^2 - D_i^1$ 。

以上是构造邻域的几种不同的插入、交换的操作, 但在算法实际应用过程中, 需要制定一定的规则以确定每一步将采用哪种操作来构造邻域、产生候选集, 并且当确定实施哪一种操作后, 还需确定具体选择哪几条路径、哪几个点来进行操作。

对于以上问题, 本研究采用如下的方法来解决: 首先, 本研究中设定侯选集的大小为 \sqrt{n} , 其中 n 为问题中的节点数。即侯选集中包含有 \sqrt{n} 个由构造邻域而产生的邻居。这 \sqrt{n} 个邻居具体的产生方法为, 首先在 or-opt 交换法、1-0 节点交换法和 1-1 节点交换法中由计算机随机选取一个方法, 确定采用的方法后, 再由计算机随机选取一或两条路径中的几个供应商点, 然后考察对其进行操作是否满足约束条件, 若满足, 则进行操作, 一个邻居生成完毕; 若不满足, 则重新随机选择一个操作方法, 并重新随机选取路径和点, 直到生成一个新的邻居为止。以此随机的方法, 逐一生成侯选集中的十个邻居, 并从中选择最好的一个考察其是否被禁忌、是否可特赦等, 以替代当前解 x^{now} 。

4.3.4. 禁忌对象及禁忌长度

对以上的邻域搜索方法, 建立以解的向量分量为禁忌对象的禁忌表, 具体如表 1 所示。

以上禁忌对象的禁忌长度均为 $t = \sqrt{n}$, 其中 n 为问题中的节点数。

4.3.5. 特赦规则

本研究的特赦规则将采用基于评价值的规则, 即当出现一个解 x^{now} 的评价值与已出现的最好解 x^{best} 的评价值相比时, $E(x^{now}) < E(x^{best})$, 则即使从 x^{best} 向 x^{now} 的变化是被禁忌的, 此时也可以解禁。

Table 1. Table of tabu objects

表 1. 禁忌对象表

交换方法	禁忌对象
Or-opt	(r, i)
1-0 交换	(r, i, r_j)
1-1 交换	(r_i, i, r_j, j)

4.3.6. 终止规则

采用双重终止规则，即当总的迭代次数达到某一充分大的数 N ，或者在某一给定的步数 L 之内目前最好解没有改进，则算法终止。

5. 算例和实验分析

在之前的 VRPTW 问题研究中，多数学者采用 Solomon 标准测试数据集对自身算法进行测试。本文所研究的需求可拆分情境下卸货时间窗约束的汽车零部件循环取货路径问题没有标准的测试数据集，因此本文同样采用 Solomon 测试数据来验证需求可拆分算法的有效性。

由于以往 Solomon 的实验中目标函数只考虑了系统形式距离的长短，因此，本文的实验同样将距离作为实验的目标函数，同时，我们假设汽车零部件的包装体积、重量等物理参数为标准统一的规格，运输所采用的车辆容量是以装载货物单位包装的数量为基本单位。

本实验采用 Matlab 语言编写程序进行模拟。硬件配置如下：CPU 采用 Intel Core i7-4720HQ @2.60 GHz；内存容量为 8 GB，内存频率为 1600 MHz。

实验将通过对比使用本文算法求解 Solomon 标准测试数据得到的最优解与目前已知的使用启发式算法得到的最佳近似解(不可拆分的情况下)及文献[14]所设计的禁忌搜索算法求得的最优解来验证本文算法的有效性。由于 Solomon 数据中没有引入卸货时间窗，本文选取各个供应商货物回到 RDC 节点的时间点上下浮动一个标准节点卸货时间来构建供应商卸货时间窗，对比结果如表 2 所示。

Table 2. The comparison of three algorithms in Solomon data

表 2. Solomon 数据三种算法最优解对比

数据集	已知最优解	文献 11 最优解	本算法最优解
R101	1645.79	1648.96	1631.66
R102	1486.12	1506.08	1467.56
R103	1292.68	1276.58	1227.96
R104	1007.24	1059.42	1076.12
R105	1377.11	1448.89	1429.87
R106	1251.98	1282.50	1262.34
R107	1104.66	1229.47	1186.70
R108	960.88	962.33	945.171
R109	1194.73	1241.21	1142.94
R110	1118.59	1212.11	1116.21
R111	1096.72	1098.37	1099.64
R112	982.14	1000.12	959.872
R201	1252.37	1236.93	1236.91
R202	1191.7	1138.91	1106.71
R203	939.54	911.87	927.68
R204	825.52	844.13	839.74
R205	994.42	1035.57	1042.08
R206	906.14	990.42	980.74
R207	893.33	909.18	886.94
R208	726.75	766.37	769.82

续表

R209	909.16	966.11	964.3
R210	939.34	950.99	950.98
R211	892.71	833.47	796.09
C101	828.94	828.94	809.04
C102	828.94	840.30	818.7
C103	828.06	834.56	835.77
C104	824.78	855.30	834.44
C105	828.94	828.94	825.55
C106	828.94	828.94	807.93
C107	828.94	828.94	800.45
C108	828.94	828.94	828.18
C109	828.94	828.94	825.79
C201	591.56	591.56	591.56
C202	591.56	591.56	592.5
C203	591.17	591.17	619.64
C204	590.60	612.51	596.1
C205	588.88	588.88	588.88
C206	588.49	588.49	588.49
C207	588.29	588.29	598
C208	588.32	588.35	585.29
RC101	1696.94	1743.32	1655.54
RC102	1554.75	1564.88	1516.21
RC103	1261.67	1303.40	1303.69
RC104	1135.48	1203.85	1217.7
RC105	1629.44	1629.84	1612.16
RC106	1424.73	1459.59	1468.7
RC107	1230.48	1282.58	1189.45
RC108	1139.82	1268.03	1123.8
RC201	1406.91	1375.94	1338.84
RC202	1367.09	1283.78	1273.63
RC203	1049.62	1142.61	1147.21
RC204	798.41	867.72	894.52
RC205	1297.19	1345.87	1370.22
RC206	1146.32	1121.40	1211.25
RC207	1061.14	1045.55	1029.95
RC208	828.14	987.41	879.17

通过观察表 2 发现, 在 56 组测试数据中, 本文算法有 33 组最优解优于文献[14]或目前已知最好非拆分解, 这充分证明了本文算法的有效性, 特别是由于文献[14]也采用了拆分的方法, 通过对比说明了本文拆分算法优于文献[14]的拆分算法。但是只对比本算法与目前已知最优非拆分算法的解就会发现, 虽然本

文算法在某些数据集上更有优势,仍有部分不及目前已知最优解,其中 R1、C1、RC1 数据集中开放时间窗较窄,因此引入卸货时间窗后对解的优化有较好的效果,29 组数据中有 24 组优于文献[14]的最优解,R2、C2、RC2 数据集对供应商的开放时间窗较宽,因此对供应商取货时间的优化不明显,在 27 组数据中有 17 组优于文献[14]的最优解。此外,Solomon 标准测试数据中设置的车辆容量较大,多数车辆在装载率较低的情况下就受限于时间窗约束返回 RDC 终止服务,会导致本算法计算时无法充分发挥需求可拆分的特点。

6. 结论

本文针对当今汽车零部件供物流的循环取货路径规划问题,引入卸货时间窗的概念,提出需求可拆分的路径规划方案,并在此情景上建立数学模型,设计了改进的启发式算法对模型进行求解,通过采用 Solomon 标准测试数据验证了所设计算法的有效性,验证了卸货时间窗情境下,本算法可以有效提升 Solomon 标准数据中大部分路径规划解得质量。特别是在原数据中时间窗约束较窄的情况下,本拆分算法的优越性显著。此外,为方便计算,本文在验证算法有效性过程中假设汽车零部件的包装体积、重量等物理参数为标准统一的规格,运输所采用的车辆容量是以装载货物单位包装的数量为基本单位。在实际情况中,这一假设并不合理,此时可以考虑以货物和车辆的体积或质量为单位进行计算。同时,本文仅考虑了单一 RDC、单一车型的情况。在后续研究中,可以加入多个 RDC、多车型的情景,进一步考察本算法的性能,在商业领域,本文引入卸货时间窗概念,准确描述当今汽车零部件入厂物流中主机厂(RDC)占据主导地位确定到货时间,各供应商根据卸货时间窗决定自身装货时间的情景,在后续研究中可以进一步探索卸货时间窗对循环取货路径规划的系统影响。

致 谢

本论文是在导师朱道立教授的悉心指导下完成的。老师渊博的专业知识,严谨的治学态度,精益求精的工作作风,诲人不倦的高尚师德,严以律己、宽以待人的崇高风范,朴实无华、平易近人的人格魅力对我影响深远。不仅是我树立了远大的学术目标、掌握了基本的研究方法,还使我明白了许多待人接物与人处事的道路。

同时我还要感谢我的同门师兄——赵磊。赵磊师兄在朱老师门下攻读博士,在算法领域上有所建树。在我遇到算法设计上的疑惑时,他总能细心的引导我找到解决问题的方法。

本论文从选题到完成,倾注了老师大量的心血,获得学长的耐心帮助。在此,谨向朱道立老师和赵磊学长表示崇高的敬意和衷心的感谢!

参考文献 (References)

- [1] Bowersox, D.J., Copper, M.B. and Closs, D.J. (2002) *Supply Chain Logistics Management* (Vol. 2). McGraw-Hill, New York.
- [2] Du, T., Wang, F.K. and Lu, P.Y. (2007) A Real-Time Vehicle-Dispatching System for Consolidating Milk Runs. *Transportation Research Part E: Logistics and Transportation Review*, **43**, 565-577. <https://doi.org/10.1016/j.tre.2006.03.001>
- [3] Nemoto, T., Hayashi, K. and Hashimoto, M. (2010) Milk-Run Logistics by Japanese Automobile Manufacturers in Thailand. *Procedia-Social and Behavioral Sciences*, **2**, 5980-5989. <https://doi.org/10.1016/j.sbspro.2010.04.012>
- [4] Satoh, I. (2008) A Formal Approach for Milk-Run Transport Logistics. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **91**, 3261-3268. <https://doi.org/10.1093/ietfec/e91-a.11.3261>
- [5] 徐秋华. Milkrun-循环取货方式在上海通用汽车的实践和应用[J]. *汽车与配件*, 2003(3): 21-24.
- [6] Savelsbergh, M.W. (1985) Local Search in Routing Problems with Time Windows. *Annals of Operations Research*, **4**, 285-305. <https://doi.org/10.1007/BF02022044>

-
- [7] Golden, B.L., Wasil, E.A., Kelly, J.P. and Chao, I.M. (1998) The Impact of Metaheuristics on Solving the Vehicle Routing Problem: Algorithms, Problem Sets, and Computational Results. In: *Fleet Management and Logistics*, Springer US, 33-56. https://doi.org/10.1007/978-1-4615-5755-5_2
- [8] Solomon, M.M. (1987) Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, **35**, 254-265. <https://doi.org/10.1287/opre.35.2.254>
- [9] Banos, R., Ortega, J., Gil, C., Márquez, A.L. and de Toro, F. (2013) A Hybrid Meta-Heuristic for Multi Objective Vehicle Routing Problems with Time Windows. *Computers & Industrial Engineering*, **65**, 286-296. <https://doi.org/10.1016/j.cie.2013.01.007>
- [10] Dror, M. and Trudeau, P. (1989) Savings by Split Delivery Routing. *Transportation Science*, **23**, 141-145. <https://doi.org/10.1287/trsc.23.2.141>
- [11] Archetti, C., Speranza, M.G. and Hertz, A. (2006) A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem. *Transportation Science*, **40**, 64-73. <https://doi.org/10.1287/trsc.1040.0103>
- [12] 孟凡超, 陆志强, 孙小明. 需求可拆分车辆路径问题的禁忌搜索算法[J]. 计算机辅助工程, 2010, 19(1): 78-83.
- [13] 熊浩, 鄢慧丽. 需求可拆分车辆路径问题的三阶段禁忌算法[J]. 系统工程理论与实践, 2015(5): 1230-1235.
- [14] Ho, S.C. and Haugland, D. (2004) A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows and Split Deliveries. *Computers & Operations Research*, **31**, 1947-1964. [https://doi.org/10.1016/S0305-0548\(03\)00155-2](https://doi.org/10.1016/S0305-0548(03)00155-2)

期刊投稿者将享受如下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: mse@hanspub.org