

公路物流智能配载的研究和装载算法设计

蓝启明, 张东 站

LAN Qiming, ZHANG Dongzhan

厦门大学 信息科学与技术学院, 福建 厦门 361000

College of Information Science and Technology, Xiamen University, Xiamen, Fujian 361000, China

LAN Qiming, ZHANG Dongzhan. Research of logistics intelligent stowage and loading algorithm design. *Computer Engineering and Applications*, 2012, 48(33): 237-243.

Abstract: According to the principles and features of logistics in highway transportation, combining expert knowledge and strategies, this paper proposes a design idea and implementation method of logistics intelligent stowage. It establishes an intelligent stowage model under multidimensional constraint condition, such as different waybill types, goods attributes, car parking spots, loading requirements. It defines terms on logistics stowage and makes rules of goods loading and uses a hybrid algorithm based on heuristic algorithm and greedy algorithm to solve packing problem. It implements the logistics intelligent stowage system under ASP.NET programming environment and imitates some data of logistics companies to test the system. The test result shows this intelligent stowage method is reasonable and feasible.

Key words: intelligent stowage; full carload; less-than carload; three-dimensional packing; greedy algorithm

摘 要: 根据公路运输中物流配载的原则和特点, 结合专家知识和策略, 提出了物流智能配载的设计思想和实现方法, 建立了在不同运单类型、货品属性、出车地点、装车要求等多维度约束条件下的智能配载模型, 给出了详细的物流配载术语定义和配载规则。提出了一种基于启发式思想和贪婪思想的混合算法解决货品装箱问题。使用 ASP.NET 开发环境, 实现物流运输智能配载系统, 模拟部分物流公司货运数据进行测试, 测试结果显示了该智能配载方法的有效性与高效性。

关键词: 智能配载; 整车; 零担; 三维装箱; 贪婪算法

文献标识码: A **中图分类号:** TP399 **doi:** 10.3778/j.issn.1002-8331.1207-0191

1 引言

21 世纪以来, 随着电子商务^[1-2]的兴起, 物流业在我国正以前所未有的速度发展。现代物流因其信息化、智能化的特点引起了业内人士的高度关注。其中, 公路物流中的智能配载在实现物流公司优化调度^[3]和减少物流运输成本环节上起到重要作用, 而货运车的配载问题是实现物流智能配载的核心。在国内, 已经有不少学者开始了货运车配载问题的研究, 并提出了相关的模型设计和解决方案。陈佳娟等^[4]提出了运输管理系统中车辆配载的专家知识模型, 王鸿鹏^[5]设计了基于知识的集装箱自动配载专家

系统, 杜嘉立^[6]等则提出了专家系统在船舶配载上的智能控制策略。

本文以物流运单为基本单位, 设计了一套完整的公路物流运输智能配载解决方案。首先提出运单分类算法, 该算法有效地将含有互斥货品的运单分开处理, 放入一个待处理集合中; 其次, 根据确定的优化策略, 结合人工选车的经验, 提出配装选车算法, 该算法从货运车辆待选集合中筛选出满足运单约束的车辆; 最后, 以国内学者张德富^[7]等在解决三维装箱问题上提出的基于块装载的启发式算法为基础, 利用贪婪算法^[8-9]的思想加以改进, 使该算法在物流

基金项目: 国家自然科学基金(No.50604012)。

作者简介: 蓝启明(1987—), 男, 硕士研究生, 主要研究方向为数据库高级应用、物流运输; 张东 站(1974—), 男, 博士, 副教授, 主要研究方向为数据挖掘技术、数据库。E-mail: blue2662@qq.com

收稿日期: 2012-07-16 **修回日期:** 2012-09-28 **文章编号:** 1002-8331(2012)33-0237-07

配载实际应用中的时间效率得以提高。该智能配载模型如图1所示。

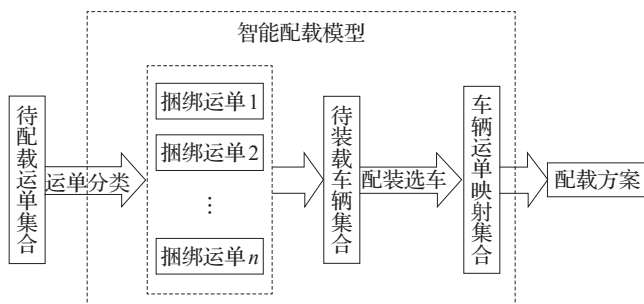


图1 智能配载模型

2 问题定义

给定一系列待装货车辆和一系列待配载运单，目标是把这些物流运单以某种策略分配给这些待装货的车辆，最终使这些运单中的待运货品合理可行地装载到这些车辆上。在考虑配载的过程中，车辆和运单之间必须满足如下3个约束条件：

(1)同一辆车内不能存在互斥的货品(如食品不能和危险品同放)。

(2)一辆车内所放置的货品总体积 S 不能超过该车辆的可装载体积 V ，即 $S \leq V$ 。

(3)车辆所在的场站 S 的运输范围需包含运单的起运地址 D ，即 $D \in S$ 。

在满足上述三个条件的基础上，配载的结果要尽可能地优化并接近人工配载的效果，优化策略主要考虑如下两个：

(1)在单次配载过程中，可能筛选出多辆满足条件的车辆，这时需要从中选出能够为物流公司减少运输成本最明显的车辆。

(2)对于需要多辆车进行装运的运单，应尽可能多的在每辆车中装载货品以使运输效益最大化，即考虑三维装箱问题^[10-11]的实际应用。

3 运单分类

有一系列的运单集合 $B = \{B_1, B_2, B_3, \dots, B_n\}$ ，车辆集合 $C = \{C_1, C_2, C_3, \dots, C_n\}$ ，货品集合 $G = \{G_1, G_2, G_3, \dots, G_n\}$ ，其中 $B_i \in B$ ，为第 i 个运单所包含的货品集合， $B_i = \{G_{i1}, G_{i2}, G_{i3}, \dots, G_{im}\}$ ， V_i 和 v_i 分别为第 i 辆车和第 i 个运单中货品的体积。运单分类算法将需要进行配载的所有初始运单进行分类。(bills)描述了一个初始运单序列，它由要进行配载的所有运单构成，运单分类算法返回一个 billList，该集合每一个元素中的运单可以捆绑在一起进行一次选车配载。在运单分类规则中，涉及到以下几个概念。

3.1 整车运单

公路物流运输中，当一个运单的货品由于重量、体积、属性的限制，需要1辆或1辆以上的货车进行装运时，该运单称为整车运单(一单多车)。整车类运单规定，即使货车中有剩余空间，也不能继续装载非此运单中的货品。所以在运单分类算法中，整车运单必须单独生成为结果集中的一个元素进行处理。

定义1(整车约束)在一次配载中，对于任意一个整车类运单，其货品总体积大于或等于任意一辆货车的承载体积，即

$$\forall B_i \in B, \exists \sum_1^n v_i \geq V_i (v_i \in B_i) \quad (1)$$

3.2 零担运单

零担运单是相对于整车运单的一个概念，如果运单中的货品不足装满一辆车，则把该类运单作为零担运单处理。一辆货车允许装运多个零担运单，但一个零担运单中的货品不能拆卸开放在多辆货车中。由于运单起运地、装车要求、货品属性等多种约束的存在，在运单分类算法中需要充分考虑不同零担类运单中的货品能否捆绑在一起进行选车配载。

定义2(零担约束)在一次配载中，对于任意一辆货车，其所配载的零担运单中的货品总体积不能大于该车的承载体积，即

$$\forall C_i \in C, \exists \sum_1^n v_i < V_i (v_i \in B_i) \quad (2)$$

3.3 运单分类算法

结合实际物流运输中的货品属性，本配载模型将货品作出如下划分：

- (1)低温冷藏品；
- (2)鲜活品(肉、鱼、水果、蔬菜等)；
- (3)危险品(化工液体、液化气体、粉粒物料、易燃易爆品等)；
- (4)普通货品。

运单分类算法考虑的约束如下：

- (1)整车运单和零担运单不能进行捆绑配载；
- (2)运单起运地不同的运单不能进行捆绑配载；
- (3)货品属性不同的运单不能进行捆绑配载。

基于以上约束设计的运单分类算法如图2所示。

4 配装选车

运单分类完成以后，开始执行对 billList 中每一个元素的选车算法。在公路物流运输中，每个物流公司都会配备各种类型的车辆，这里设车辆集合为 $C = \{C_1, C_2, C_3, \dots, C_n\}$ 。配单员会根据货品属性的不

```

algorithm BillsPartition
input(bills)
billList:={}
for i:=0 to bills.length do
    if(GetBillType(bills[i])==1) then
        add bills[i] to billList
        bills[i]:=0
    endif
endfor
for i:=0 to bills.length do
    billTemp:=empty
    if(bills[i]!=0) then
        for j:=i+1 to bills.length do
            if(bills[j]!=0) then
                if(isSameBill(billArray[i],billArray[j])) then
                    merge billArray[i] and billArray[j] to billTemp
                endif
            endif
        endfor
        if(billTemp!={}) then
            add billTemp to billList
        endif
    endif
endfor
for i:=0 to bill.length do
    if(bill[i]!=0) then
        add bill[i] to billList
    endif
endfor
return billList
    
```

图2 运单分类算法

同选择特定的车辆进行货品装运。在实际配载中,配单员还需考虑运单的装车要求、货品体积、运单起运地、车辆出车地点等多种约束条件。所以,在一般情况下很难得到最优的选车结果。配载模型依据人工选车的基本策略,尽可能地将选车结果的效益最大化。

4.1 选车的专家知识和策略

(1) 货品属性约束

货品属性约束规定,每种类别的货品只能使用特定的车辆进行装运。具体约束规则如表1所示。

表1中每种类型的货车只能装运货品栏中打“√”的货品,例如厢式货车只能装运普通货物和鲜活货品,不能装运危险品、低温冷藏品。

(2) 装车要求约束

配单员在配单时,主要会考虑车型、所属车队、车长、车重、运输线路等五个装车要求(装车要求可根据物流公司实际情况动态添加)。假设装车要求集合为 $P = \{P_1, P_2, P_3, \dots, P_n\}$, 在选车时,车辆必须满足

表1 不同类型车辆可以装运的货品

	普通货物	危险品	鲜活	低温冷藏
厢式货车	√	×	√	×
油罐车	×	√	×	×
仓栅车	√	×	×	×
粉粒物料运输车	×	√	×	×
客货两用车	√	×	√	×
冷藏车	×	×	×	√
液化气体运输车	×	√	×	×
挂车	√	×	√	×
平板车	√	×	×	×
化工液体运输车	×	√	×	×
微车	√	×	√	×
自卸车	√	×	√	×

集合 P 中的每个 P_i 。当然,装车要求也可以为空集,这时默认所有车辆都满足装车要求。

(3) 货品体积约束

在单次配载过程中,货品的总体积为 billList 中一个元素包含的运单所对应货品的体积总和 V_n 。在进行配装选车时,所选车辆的容量 V_c 必须大于该次配载的货品体积总和,即 $V_c \geq V_n$ 。

在进行配装选车时,并不需要考虑货品的装箱问题,而仅需满足货品体积约束即可。这样可以筛选出更多满足条件的车辆,使接下来的装箱问题有更多的选择空间。

(4) 地址约束

在一个物流公司经营范围的城市内,至少有一个出车场站,每个场站都有对应货运车。配装选车算法考虑的运单都有一个起运地址。地址约束规定,对于运单的起运地址,算法将筛选出和这个地址所属城区相匹配的车辆,若结果为空,则放宽地址约束,筛选出和这个地址所在城市相匹配的车辆。

4.2 选车排序算法设计

经过以上几个约束过滤出的车辆即为运单可配载车辆,将这些车辆放入集合 satCarList 中。在最终确定选择 satCarList 中哪些车进行装配时,主要考虑车辆利用率和容量比两个因素。

设计算法前,首先给出以下几个定义:

定义3(利用率) 待配运单完成配载后,货品与车辆的体积(重量)比。由此定义可得:

$$\text{体积利用率}(V_UseRate) = \frac{\sum \text{待配运单体积}}{\text{车辆额定承载体积}} \quad (3)$$

$$\text{载重利用率}(W_UseRate) = \frac{\sum \text{待配运单重量}}{\text{车辆额定承载重量}} \quad (4)$$

定义4

$$\Delta \text{容量比}(C_Ratio) = |\text{配载容量比} - \text{车辆容量比}| \quad (5)$$

定义5

$$\text{配载容量比}(m^3/t) = \frac{\sum \text{待配运单体积}}{\sum \text{待配运单重量}} \quad (6)$$

定义6

$$\text{车辆容量比}(m^3/t) = \frac{\text{车辆额定承载体积}}{\text{额定承载重量}} \quad (7)$$

在对可配载车辆集合 satCar 排序时, 主要考虑车辆体积利用率 $V_UseRate$ 、车辆载重利用率 $W_UseRate$ 以及容量比 C_Ratio 三个因素, 这里引入权值的概念, 对三个因素赋予不同的权值 W_i , 经过计算得到每辆车的权值 CW 。

定义7

$$\text{车辆权值比 } CW = (V_UseRate \times W1 + W_UseRate \times W2 + C_Ratio \times W3) / (W1 + W2 + W3) \quad (8)$$

根据以上定义得到选车排序算法如图3所示。

```

algorithm CarSort
input (satCarList, billList, W1, W2, W3)
V_UseRate := {}
W_UseRate := {}
C_Ratio := {}
CW := {}
for each car in satCarList do
    V_UseRate[car] := billList.Volume/car.Volume
    W_UseRate[car] := billList.Weight/car.Weight
    C_Ratio[car] := Abs(car.Volume/car.Weight - billList.Volume/billList.Weight)
    CW[car] := (V_UseRate[car] × W1 + W_UseRate[car] × W2 + C_Ratio[car] × W3) / (W1 + W2 + W3)
endfor
QuickSort(satCarList, CW)

```

图3 选车排序算法

算法中, QuickSort 利用快速排序的思想, 将 satCarList 中的车辆按照车辆权值比 CW 从大到小进行排序。

4.3 配装选车算法

配装选车算法维护一个初始车辆集合 initCar 和一个可满足车辆集合 satCar , 算法遍历要配载的运单集 billList , CarSelect 执行对单一运单可满足车辆的选择, 选择的约束条件如4.1节中所述。每个运单的车辆选择结果保存在过滤集 filterCar 中, 通过取 initCar 和 filterCar 的交集找到可满足所有待配运单的车辆, 即 $\text{satCar} = F_1 \cap F_2 \cap F_3 \cap \dots \cap F_n$ (其中 F_i 为运单集合中每个运单可满足的车辆集), 完整算法如图4所示。

设 satCar 中有 n 辆车, 如果在一次配载中有车辆数量限制, 即配载车辆不能大于 m ($m < n$), 那么根据

```

algorithm GenCarList
input (billList)
initCar := {}
filterCar := {}
satCar := {}
init := 0
for each bill in billList do
    if init=0 then
        initCar := CarSelect(bill, billList.TotalVolume)
        satCar := CarSelect(bill, billList.TotalVolume)
        init := 1
    else then
        satCar := {}
        filterCar := CarSelect(bill, billList.TotalVolume)
        for each car in filterCar do
            if initCar contains car then
                add car to satCar
            endif
        endfor
        initCar := satCar
    endif
endfor
return satCar

```

图4 配装选车算法

选车排序算法, 在对可满足车辆集 satCar 排序后, 删除列表中最后 $n - m$ 个元素。

5 三维装箱问题的研究

三维装箱问题在工业和运输业领域有着广泛应用, 在物流运输的装车过程中, 装车员主要依据经验对货品进行摆放和安置。由于三维装箱是一个 NP-hard 问题^[12], 所以用确定性的方法精确求解是不现实的, 因此启发式的方法就体现出在解决三维装箱问题上的优越性。本配载模型在解决货品的装箱问题上, 首先用贪婪算法的思想在每次可选择车辆中选取权值最大的一辆车; 然后利用基于块装载的启发式算法对这辆车进行装填, 得到货品的装填集。

5.1 基于贪婪思想的装箱算法

贪婪算法是一种追求局部最优解^[13]的算法, 在对问题求解时, 算法做出的选择都是当前看来最好的解。也就是说, 不从全局上考虑, 仅获得局部意义上的最优解。贪婪算法不能对所有问题都得到全局最优解, 但是在解决很多问题上都能得到很好的效果。

贪婪算法采用自顶向下的方法, 省去为了寻求最优解而搜索所有可能所要花费的大量时间, 每次做出贪心选择后, 贪婪算法都将问题进一步简化为一个规模更小的子问题, 直到问题结束。在对货品

进行装箱时,对于那些无法用一辆车装满的货品,贪心思想每次选择出按车辆体积为权重进行排序后所得到的最大权值的车。具体的装载策略使用基于块装载的基础启发式算法,这样就有可能用最少的车辆完成货品的装载,为物流公司节省运输成本。

基于贪婪思想的装车算法可以描述为:有一系列可选车辆集合 $carList = \{C_1, C_2, C_3, \dots, C_n\}$, 待配载运单的剩余货品集合 $restGoods = \{G_1, G_2, G_3, \dots, G_n\}$, $C_i \in C$ 为第 i 辆车中装载货物的集合, $C_i = \{G_{i1}, G_{i2}, G_{i3}, \dots, G_{im}\}$, 车辆选择集 $P = \{P_1, P_2, P_3, \dots, P_n\}$ 对应 $carList$, 当 C_i 被选择时, P_i 为 1, 否则 P_i 为 0, V_i 为 C_i 车辆的体积, $restVolume$ 为剩余货品的总体积。

目标函数为 $\min \sum_1^n P_i$, 约束条件为:

$$\forall V_i \in V, \exists V_i < restVolume, \sum_1^n C_i = \sum_1^n G_i$$

算法描述如图 5 所示。

```

algorithm GenGreedyCar
input(billList, carList)
CarSort(carList, billList, 1, 0, 0)
restGoods:={}
restVolume:=0
count:=0
for each bill in billList do
    restVolume=carList[count].Volume
    restGoods=bill.Goods
    if bill.Type is fullLoad then
        while(!isFull(restGoods)) then
            execute BasicHeuristic(carList[count], restGoods, restVolume)
            GenLoadInfo(bill, carList[count])
            reduce the goods in restGoods
            count++
        else then
            if bill.Volume>carList[count].Volume then
                restVolume:=carList[count++].Volume
            endif
            GenLoadInfo(bill, carList[count])
            restVolume:=restVolume - bill.Volume
        endif
    endif
endfor

```

图 5 贪婪装箱算法

针对整车类运单,执行 BasicHeuristic(块装载启发式算法),该算法将货品划分为长方体块,然后将其填充进容器(即车辆)中,返回填充的结果集。GenLoadInfo 生成货品填充明细,其中货品按照其重量体积比(容重)从大到小排序,同时建立货品和配载车辆的映射关系。

5.2 货品装车策略

装箱算法执行后,接下来需要考虑货品装车顺序问题。该问题解决如何将货品合理有序地装进车厢内。具体可按车厢空间,分为装车前后顺序和上下顺序两方面。首先给出该问题涉及到的几个概念:

(1) 车厢

车厢是问题中被装载的长方体,用 lx, ly, lz 三个域来描述它三条边的长度。

(2) 运单地址

运单中的地址有两个。用 start 表示运单的起运地, destination 表示运单的目的地, distance 描述 start 到 destination 的距离。运单地址决定货品在车厢中摆放的前后顺序。

(3) 货品优先级

货品优先级 priority 分为重质和轻浮两种,分别用 high 和 low 表示。priority 决定货品在车厢中摆放的上下顺序。

在货品装车时,主要使用如下两种策略:

(1) 根据运单地址,将先卸货的货品摆放在车厢前面,后卸货的货品摆放在车厢后面。

(2) 货品上下摆放时,根据重不压轻,大不压小的基本装车原则,将货品优先级为 low 的货品放在货品优先级为 high 的货品上面,并且保证货品在堆放后的重心平衡。

图 6 描述了基于以上装车策略的装车单生成算法。

```

algorithm GenGoodsList
input(billList, goodsList)
desList:={}
dis[0..n-1]
i:=0
if bill.Type is fullLoad then
    load goods from bottom up order by priority
else
    for each bill in billList do
        if bill.destination is not in desList then
            add bill.destination to desList
            dis[i]:=compute distance from start to bill.destination
            i++
        else
            merge the goods to the destination
        endif
    endfor
    Sort bills order by dis
    cut the freight car into i blocks
    put goods into the blocks according to dis
        load goods in each block from bottom up by priority
    endif
endfor

```

图 6 装车单生成算法

根据运单定义,若货运车装运整车类运单,则该

车目的地只有一个,所以不需要考虑货品的前后装车顺序。对于该类运单,装车单生成算法 GenGoods-List 直接按照货品优先级进行摆放。若货运车装运零担类运单,GenGoodsList 首先得到运单集中的所有目的地集合 *desList*;然后按照起始地到目的地的距离 *dis[0..n-1]*,将运单前后摆放位置进行排序并生成相应摆放块 block;最后将每个 block 中的货品按照货品优先级摆放。图 7 展示了基于该算法摆放后的装车效果示意图。

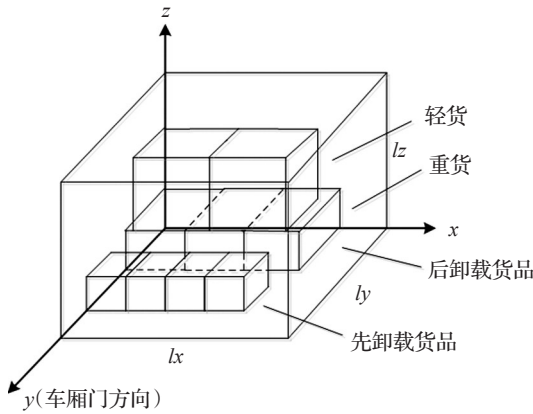


图7 货品装车效果简图

6 智能配载的实现和实验

6.1 智能配载算法

完整的智能配载算法如图 8 所示。

```

algorithm IntelligentLoad
input(bills, W1, W2, W3)
carList:={ }
billList:=BillsPartition(bills)
for each bills in billList do
    carList:=GenCarList(bills)
    CarSort(carList, bills, W1, W2, W3)
    if carList.length>0 then
        if carList[0].Volume<bills.Volume then
            GenGreedyCar(bills, carList)
        else then
            GenLoadInfo(bills, carList[0])
        endif
    endif
endfor
    
```

图8 智能配载算法

6.2 实验测试

使用 ASP.NET 开发环境及 C# 编程语言实现智能配载模型和相关装载算法,测试数据存储在 SQL SERVER 2005 数据库中。实验环境: Pentium® Dual-CPU E5400 @ 2.7 GHz, 内存 4 GB, 500 GB 硬盘, Windows Server 2003 操作系统以及 Visual Studio 2008。

6.2.1 配载效果测试

提取某物流公司相关数据进行模拟测试,表 2 记录该物流公司所有车辆数据,待配载运单记录如表 3 所示(运单对应货品在此没有逐项列出,通过运单分类后,已排除了货品互斥的情况),最终的配载结果如表 4 所示。

表2 车辆列表

车牌号码	车辆类型	出车区域	承载体积/m ³	承载重量/t
闽DY1347	客货两用车	厦门站	2.0	2.0
闽DY2408	客货两用车	思明站	7.6	1.2
闽DY3368	厢式货车	厦门站	8.0	1.8
闽DY4009	厢式货车	思明站	12.0	2.5
闽DY5366	厢式货车	福州站	30.0	5.0
闽DY6865	冷藏车	厦门站	77.0	10.0
闽DY7719	厢式货车	福州站	12.0	2.5
闽DY8900	冷藏车	厦门站	45.0	8.0
闽DY9191	平板车	泉州站	96.0	40.0
闽DY0106	平板车	泉州站	100.0	25.0

表3 运单列表

运单编号	类别	始发地	目的地	体积/m ³	重量/t	货品性质
YD1205130001	零担	厦门市	福州市	1.00	1.2	普通货品
YD1205130002	整车	厦门市	杭州市	60.00	10.0	低温冷冻
YD1205130003	整车	福州市	杭州市	40.20	17.2	普通货品
YD1205130004	零担	厦门市	泉州市	1.50	0.4	普通货品
YD1205130005	零担	厦门市	泉州市	3.00	0.8	普通货品
YD1205130006	零担	厦门市	上海市	1.50	0.8	普通货品
YD1205130007	零担	厦门市	上海市	5.00	4.0	普通货品
YD1205130008	零担	厦门市	上海市	0.25	0.3	普通货品
YD1205130009	零担	厦门市	杭州市	6.50	6.4	普通货品

表4 智能配载实验结果

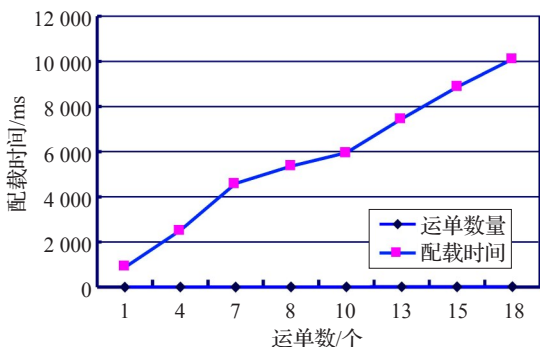
作业单号	车牌号	运单集	承载体积/m ³	配载体积/m ³
W1205130001	闽DY5366	YD1205130003	30	29.00
W1205130002	闽DY7719	YD1205130003	12	9.00
W1205130003	闽DY6865	YD1205130002	77	60.00
W1205130004	闽DY4009	YD1205130009 YD1205130007	12	11.50
W1205130005	闽DY3368	YD1205130005 YD1205130006 YD1205130008	8	7.25

6.2.2 配载时间测试

假设如果初始运单集在执行运单分类算法后只生成一种捆绑运单(即没有互斥运单),这样的配载问题为同构配载问题,有少数几种捆绑运单且每种捆绑运单中运单数量较多的配载问题为弱异构配载问题。强异构配载问题则有许多种捆绑运单且每种捆绑运单中的运单数量较少。实验测试主要针对异

构问题,进行配载时间测试。

针对异构配载问题,不同数量运单智能配载时间测试如图9所示。



注:捆绑运单种类为:1,2,2,3,4,5,6,7。

图9 智能配载时间测试

7 结束语和今后工作

针对公路物流运输中的整车和零担运单,本文给出了一套智能配载解决方案,实现了从运单分类、配装选车到货品装箱整个配载过程的一键式智能化配载。根据实际的测试结果可以看到,该配载方法可基本满足国内大部分物流公司在公路运输中的货物配载需求,减少配单员在处理物流配载问题上的工作时间,提高配载效率,并以专家知识和策略优化配载结果。同时,本文提供的智能配载模型和算法也有助于今后进一步开展对铁路物流运输和船舶物流运输的研究。接下来,也会根据实际测试数据,继续跟踪和改进该智能配载模型。

(上接236页)

但从总体上看,本文提出的三段式粒子群能够较好地解决车辆的调度问题。该算法在实验中表现出较好的寻优能力,体现了算法的有效性。

4 结束语

本文研究的车辆调度是基于时间最短、路径最短的调度问题。其实际应用背景广泛,如物资应急调度、消防调度等各种车辆组合调度等。通过引入整数编码方式,解决了连续粒子群算法离散化过程中编码解码的问题,及解码后粒子有效率低等问题。算法中避免了学习因子等参数,但又重新引入了分段比例参数,其对算法的收敛速度和寻优效果有一定的影响,随着规模的扩大,比例参数的确定将是本文继续研究的方向。

参考文献:

[1] Kennedy J, Eberhart R. Particle Swarm Optimization[C]//IEEE Int Conf on Neural NetWrks, Perth, Australia, 1995:

参考文献:

[1] 魏广兰. B2C 电子商务物流管理发展趋向浅析[J]. 科技信息, 2009(19).
 [2] 刘志, 赵珣. 国内 C2C 电子商务中的物流模式探讨[J]. 科技信息, 2007(19): 216-217.
 [3] Lambert, Stock, Ellram. Fundamentals of logistics management[M]. America: McGraw-Hill, 2002: 216-219.
 [4] 陈佳娟, 王云鹏, 纪寿文. 运输管理信息系统中车辆配载研究[J]. 公路交通科技, 2004, 21(12): 138-140.
 [5] 王鸿鹏. 基于知识的集装箱船自动配载专家系统[J]. 上海海运学院学报, 2002, 23(1): 46-49.
 [6] 杜嘉立, 杨盐生, 郑云峰. 基于专家系统的船舶配载智能控制策略[J]. 中国航海, 2005(3): 50-53.
 [7] 张德富, 彭煜, 朱文兴, 等. 求解三维装箱问题的混合模拟退火算法[J]. 计算机学报, 2009, 32(11): 2147-2156.
 [8] 官国顺. 贪心算法在 P 类问题求解中的应用[J]. 电脑知识与技术, 2011(2): 198-200.
 [9] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms[M]. Beijing, China: China Machine Press, 2007: 123-146.
 [10] Scheithauer G. Algorithms for the container loading problem[J]. Operations Research Proceedings, 1991: 445-452.
 [11] 陈泽爽, 何健明, 林遼. 多约束三维装箱问题的混合遗传算法[J]. 现代计算机: 专业版, 2011(Z1): 16-21.
 [12] Jungnickel D. Graphs, networks and algorithms[M]. German: Springer, 2008: 49-52.
 [13] 张德富. 算法设计与分析[M]. 北京: 国防工业出版社, 2007: 5-6.

1942-1948.

[2] Kennedy J, Eberhart R C. Discrete binary version of the particle swarm algorithm[C]//Proc of the IEEE International Conference on System, Man, Cybernetics. Orlando: IEEE Press, 1997: 4104-4108.
 [3] Wang Wanliang, Wu Bin. Particle Swarm Optimization for open vehicle routing problem[C]//Lecture Notes in Artificial Intelligence, 2006, 4114: 999-1007.
 [4] 宁正元, 林大辉, 李丽珊, 等. 置换流水车间调度问题的离散粒子群优化算法[J]. 集美大学学报: 自然科学版, 2008, 13(2): 97-101.
 [5] 卢冰原, 何力, 程八一. 具有模糊时间窗的有容积约束车辆调度优化问题研究[J]. 计算机应用研究, 2011, 28(9).
 [6] Maurice C. Discrete Particle Swarm Optimization[M]//New Optimization Techniques in Engineering. [S.l.]: Springer Verlag, 2004: 219-240.
 [7] 田翠华. 算法设计与分析[M]. 北京: 冶金工业出版社, 2007: 74-81, 88-103.
 [8] 王桂平, 王衍, 任佳辰. 图论算法理论、实现及应用[M]. 北京: 北京大学出版社, 2011: 180-197.