

文章编号:1006-2467(2020)02-0126-09

DOI: 10.16183/j.cnki.jsjtu.2020.02.003

# 自动存取系统多载量轨道小车避碰调度方法

马昌谱<sup>1,2</sup>, 周炳海<sup>1</sup>

(1. 同济大学 机械与能源工程学院, 上海 201804; 2. 桂林航天工业学院 管理学院, 广西 桂林 541004)

**摘要:** 针对自动存取系统(AS/RS)货物装卸顺序约束和轨道小车(RGV)潜在碰撞的问题,引入装卸顺序冲突和碰撞冲突的概念对问题进行描述,以最小化任务总完工时间为目标,建立了消除冲突的数学规划模型. 针对小规模问题,应用 CPLEX 获取其最优解;对于中大规模问题,构建了改进型和声搜索算法. 在算法设计中,通过拆分、合并和声记忆库加快搜索速度,并融入 4 种变邻域搜索策略和 1 种变异操作以克服基本和声搜索算法早熟收敛、易陷入局部最优等不足. 仿真实验与其他算法的对比结果表明,所构建算法具有较高的求解质量和良好的收敛性能,能提高 AS/RS 的运行速度和实用性.

**关键词:** 自动存取系统; 避碰调度; 改进型和声搜索算法; 装卸顺序约束

中图分类号: TP 391

文献标志码: A

## Scheduling of Multi-Load Rail Guided Vehicles in AS/RS with Collision Avoidance Constrains

MA Changpu<sup>1,2</sup>, ZHOU Binghai<sup>1</sup>

(1. School of Mechanical Engineering, Tongji University, Shanghai 201804, China;

2. School of Management, Guilin University of Aerospace Technology, Guilin 541004, Guangxi, China)

**Abstract:** To deal with the problems of cargo loading/unloading sequence constraint and impending collision between rail guided vehicles (RGVs) in automated storage and retrieval system (AS/RS), the concepts of the conflicts of cargo loading/unloading sequence and collision are introduced. And a mathematical programming model considering the characteristics of eliminating the conflicts is established to minimize the total complete time. CPLEX is employed to optimally solve the small-scale problems, and a modified harmony search(MHS) algorithm is constructed for medium to large scale problems. In the algorithm design, the harmony memory is repeatedly divided and regrouped to accelerate the searching process, and four variable neighborhood search strategies and a mutation operation are integrated to overcome the deficiencies of the basic harmony search algorithm, such as limited search depth and tendencies to trap into local optimum. Simulation experiments of the MHS algorithm are compared with other evolution algorithms, and the simulation results show that the algorithm has high solution quality and good convergence performance, and can improve the running speed and the practicality of the AS/RS.

**Key words:** automated storage and retrieval system (AS/RS); collision avoidance scheduling; modified harmony search (MHS) algorithm; loading/unloading sequence constraint

收稿日期: 2018-08-03

基金项目: 国家自然科学基金(71471135), 国家社会科学基金(14XJY009), 广西教育厅科研基金(KY2015YB349)资助项目

作者简介: 马昌谱(1979-), 男, 贵州省普安县人, 博士生, 讲师, 研究方向为智能优化和生产系统建模和调度.

通信作者: 周炳海, 男, 教授, 博士生导师, 电话(Tel.): 18917009179; E-mail: bhzhou@tongji.edu.cn.

轨道小车(RGV)被广泛应用于自动化存取系统(AS/RS)<sup>[1]</sup>.作为AS/RS的主要存取部件,一次能够搬运多个物品的RGV,即多载量RGV能提高系统的作业能力,其作业效率直接影响系统的吞吐率,因此,RGV的合理调度对系统性能有着非常重要的影响.在多车AS/RS中,RGV的调度不仅要考虑多货物同时载运的装卸顺序约束,还须考虑多车之间的碰撞问题,这是RGV调度的难点.

目前,关于RGV的研究主要聚焦于其设计、控制、派遣规则及路径选择等方面.Zheng等<sup>[2]</sup>设计了环形共轨RGV,并将其应用于分载系统.Lee等<sup>[3]</sup>构建了AS/RS仿真模型,确定了RGV的最优数量和导出系统吞吐率最大化策略.Lee和Chen等<sup>[4-5]</sup>研究了RGV的调度和系统控制问题,设定RGV的分派规则,并用仿真实验对其规则做了评价.Dotoli等<sup>[6]</sup>应用着色Petri网,分析货物搬运系统并建立模型,提出控制策略,解除环轨RGV的死锁问题.Liu等<sup>[7]</sup>使用仿真实验,提出双RGV系统的两种作业策略,并分析了两者的优劣.RGV的调度方面,Kung等<sup>[8]</sup>将订单分为不同的集群,按照规则将订单集群序列分配给同轨运行的RGV.Gao等<sup>[9]</sup>构建了在线调度算法,用于调度同轨RGV,以实例证明其可行性.

纵观上述文献,系统中的RGV无论是一辆还是多辆,每次只能搬运单个物品,属于单载量RGV.针对多载量RGV的调度问题,目前仅有Hu等<sup>[10]</sup>对其作了研究,并考虑了货物装卸顺序约束,应用滚动时域法对RGV作了最优路径规划.

但是,目前尚未发现其他关于同时考虑货物装卸顺序约束和同轨双车避碰调度问题的研究.为此,本文将创新引入I类和II类冲突的概念,研究多载量RGV货物装卸顺序与同轨双车路径规划相结合的问题,以货物搬运总时间最小为优化目标,建立解除双冲突的决策模型.针对小规模问题,应用CPLEX求其最优解,并构造改进型和声搜索算法获取中大规模的全局满意解.

1 问题描述

如图1所示,AS/RS由两辆RGV( $V_1, V_2$ )共享同一直线轨道,RGV和货架都安装了滚轮滑道,方便货物的自动存取.系统由RGV按生产计划载运货物至对应货架,以满足生产需求.调度开始前, $V_1$ 和 $V_2$ 分别位于系统的最左端和最右端,调度开始时,两辆RGV分别从各自的端点启程,完成所分配的任务后,返回至各自的出发点.在此,重点需要

解决两个问题:① RGV装卸货物的先后顺序问题,即解决多件货物在RGV内部的死锁问题,此死锁是指已经装载在RGV上的货物,相互阻挡了对方的卸载通道,以致货物被迫处于停滞等待状态;② RGV之间的碰撞问题,即确定避碰情况下的最佳取送路径.

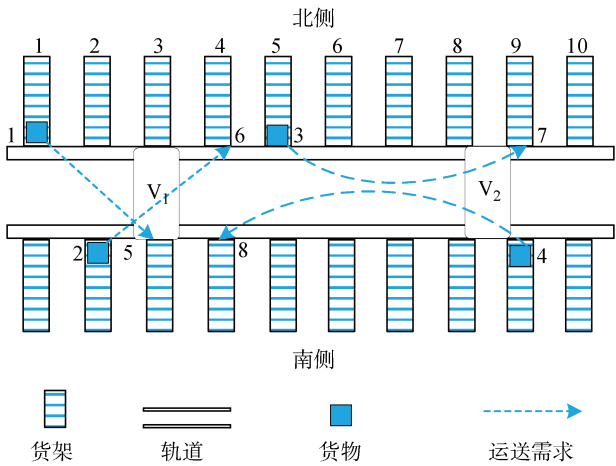


图1 以RGV为存取部件的AS/RS  
Fig.1 Configuration of AS/RS with RGV as storage and retrieval component

为有效描述该存取系统,进行如下基本假设:① RGV可同时搬运多个货物,且车上货物只能顺序装卸;② RGV装载和卸载单位货物的时间相等且为常数;③ RGV不能相互跨越,也不能发生碰撞;④ RGV速度恒定,且两辆RGV型号相同;⑤ 同一轨道位置对应的两对称货架同时只能被一辆RGV访问;⑥ 每个货物的源点和终点已知;⑦ RGV一旦装载了某件货物,到达终点之前,一件货物只能由此辆RGV完成整个搬运过程.

为深入分析问题,对部分参数描述如下:假设系统中有取货任务 $r_i$ 和 $r_j$ ( $i, j$ 为任务标号),对应的送货任务为 $r_{i+n}$ 和 $r_{j+n}$ ( $n$ 为任务总量); $P$ 和 $D$ 分别表示取货任务标号集合和送货任务标号集合,即 $i \in P, i+n \in D, V' = P \cup D$ ;四元组 $r_i = (\alpha, p, \beta, d)$ 表示一个完整任务信息,其中,前两位是源点信息, $\alpha = 1$ 为北侧, $\alpha = 2$ 为南侧, $p$ 为任务 $r_i$ 的源点所在的货架编号,后两位是终点信息, $\beta = 1$ 为北侧, $\beta = 2$ 为南侧, $d$ 表示 $r_i$ 的终点所在的货架编号.如图1所示, $r_1 = (1, 1, 2, 3)$ 表示1号任务的源点为北侧1号货架,终点为南侧3号货架.令 $K$ 为RGV集合, $k \in K, K = \{1, 2\}, k = 1, k = 2$ 分别表示左端和右端的RGV; $x_{ij}^k = 1$ 表示 $V_k$ 先取货物 $i$ ,再取货物 $j$ ,否则为0,用弧 $(i, j)$ 表示,即 $V_k$ 遍历弧 $(i, j)$ ,其弧集用

$A'$  表示;  $r_N, r_S$  分别表示北侧和南侧货架产生任务的集合.

根据货物的源点和终点不同,有4种货物路径类型:① 源点在北侧、终点也在北侧,表示为  $E^1 \triangleq \{i \in P: r_i \in r_N, r_{i+n} \in r_N\}$ ; ② 源点在南侧、终点也在南侧,即  $E^2 \triangleq \{i \in P: r_i \in r_S, r_{i+n} \in r_S\}$ ; ③ 源点在北侧、终点在南侧,即  $E^3 \triangleq \{i \in P: r_i \in r_N, r_{i+n} \in r_S\}$ ; ④ 源点在南侧、终点在北侧,即  $E^4 \triangleq \{i \in P: r_i \in r_S, r_{i+n} \in r_N\}$ . 根据以上描述,给出如下定义和定理:

**定义1** 当  $\forall i \in E^3, j \in E^4$  或  $\forall i \in E^4, j \in E^3$ , 且  $x_{ij}^k = 1$  时,会由于货物装卸顺序约束而造成死锁,称此情况为 I 类冲突.

**定义2** 对  $V_k$  有路径集合  $H^k = \{h_{k,1}, h_{k,2}, \dots, h_{k,\pi}\}$ , 其中  $h_{k,\pi} = p_{k,\pi}^1 p_{k,\pi}^2 \dots p_{k,\pi}^m p_{k,\pi}^m$  表示  $V_k$  在第  $\pi$  条路径上的第  $m$  个位置. 对于  $\forall t \in T, T$  表示离散时刻集合,若  $p_{k_1,i}^m(t) \geq p_{k_2,j}^m(t)$ , 即任意两辆 RGV 发生碰撞,则称此情况为 II 类冲突.

**定理1**  $\forall t \in T$ , 当  $V_k$  同时取送货物组合  $j \in E^3$  和  $l \in E^4$  或  $j \in E^4$  和  $l \in E^3$  时,发生 I 类冲突.

**证明** 对  $j \in E^3, V_k$  将  $r_j$  装载后在车内的移动方向是  $1 \rightarrow 2$ , 若再装载  $r_l, l \in E^4$ , 其移动方向为  $2 \rightarrow 1$ , 若将  $r_j, r_l$  形成组合搬运,根据定义1,  $r_j$  和  $r_l$  彼此阻碍了对方的目的去向,形成死锁, I 类冲突发生.

**推论1**  $\forall i \in E^4, j \in E^1$ , 或者  $\forall i \in E^3, j \in E^2$ , 若  $x_{ij}^k + x_{ji}^k + x_{i,i+n}^k + x_{i+n,j+n}^k > 1$ , 则发生 I 类冲突.

**证明** 情形1, 当  $x_{ij}^k + x_{ji}^k > 1$  时,显然是 I 类冲突; 情形2, 若  $x_{ij}^k = 1, x_{i,i+n}^k = 1$ , 则  $V_k$  的运行路径为  $i \rightarrow j \rightarrow i \rightarrow i+n$ , 即  $V_k$  是先装载  $r_i$ , 再装载  $r_j$ , 然后卸载  $r_i$ , 此时  $r_j$  仍在小车上, 由  $j \in E^1$  及 RGV 装载的顺序可知,  $r_j$  会阻挡  $r_i$  的行进路径, 故形成 I 类冲突; 情形3,  $x_{ij}^k = 1, x_{i+n,j+n}^k = 1$ ; 情形4,  $x_{ji}^k = 1, x_{i,i+n}^k = 1$ ; 情形5,  $x_{ji}^k = 1, x_{i+n,i+n}^k = 1$ ; 情形6,  $x_{i,i+n}^k = 1, x_{i+n,i+n}^k = 1$ . 情形3~6的分析类同情形2. 综上,任意两项为1的组合,都会造成 I 类冲突.

**推论2**  $\forall i \in E^1 \cup E^4, j \in E^1/\{i\}$ , 或  $\forall i \in E^2 \cup E^3, j \in E^2/\{i\}$ , 若  $x_{ij}^k + x_{i+n,j+n}^k + x_{j+n,i}^k + x_{i+n,j}^k > 1$ , 则发生 I 类冲突.

**证明** 先证明  $i \in E^1, j \in E^1/\{i\}$  的情形, 对于  $x_{ij}^k + x_{i+n,j+n}^k + x_{j+n,i}^k > 1$ , 若  $x_{ij}^k = 1, x_{i+n,j+n}^k = 1$ , 表明  $V_k$  在取货端装载顺序是  $i \rightarrow j$ , 在卸载端顺序也是  $i \rightarrow j$ ,  $r_j$  阻挡了  $r_i$  的滑行路径; 对于  $x_{j+n,i}^k = 1$ , 隐

含的前提是  $V_k$  已经装载了货物  $j$ , 这与  $x_{ij}^k = 1$  本身就构成冲突; 同理可证, 任意两项为1的组合, 都构成 I 类冲突. 同理  $\exists x_{ij}^k + x_{i+n,j+n}^k + x_{i+n,j}^k > 1$  时, 构成 I 类冲突; 将两式合并, 即得题设.  $i \in E^4, j \in E^1/\{i\}$  的情形, 证明过程同  $i \in E^1, j \in E^1/\{i\}$  的情形.

**推论3**  $\forall i \in E^4, j \in E^4/\{i\}$ , 或  $\forall i \in E^3, j \in E^3/\{i\}$ , 若  $x_{ji}^k + x_{i,i+n}^k + x_{i+n,j+n}^k > 1$ , 则发生 I 类冲突.

原理与  $i \in E^1, j \in E^1/\{i\}$  类似, 证明略.

**推论4**  $\forall i \in E^1 \cup E^4, j \in E^4/\{i\}$ , 或  $\forall i \in E^2 \cup E^3, j \in E^3/\{i\}$ , 若  $x_{ij}^k + x_{i+n,j+n}^k + x_{j+n,i}^k + x_{i+n,j}^k > 1$ , 则发生 I 类冲突.

证明过程同推论2, 即先证明  $x_{ij}^k + x_{j+n,i+n}^k + x_{j+n,i}^k > 1$  时, I 类冲突成立, 然后再证明  $x_{ij}^k + x_{j+n,i+n}^k + x_{i+n,j}^k > 1$  成立, 将两式合并即可得题设, 过程略.

## 2 数学模型

根据以上定义、定理及描述, 引入以下参数及变量, 建立数学模型:  $G$  为有向图,  $G = (V, A)$ ;  $V = \{0\} \cup V' \cup \{2n+1\}$ ,  $\{0\}$  和  $\{2n+1\}$  分别为 RGV 的虚拟起点和终点;  $A$  为图  $G$  的弧集;  $k$  为 RGV 集合,  $k \in K, K = \{1, 2\}, k=1$  为轨道左侧的 RGV,  $k=2$  为轨道右侧的 RGV;  $W$  为单侧货架集合, 从左到右递增编号,  $W = \{1, 2, \dots, |W|\}$ ;  $R$  为任务集合,  $R = \{r_1, r_2, \dots, r_n\}$ ;  $p_i$  为  $r_i$  的源点位置,  $i \in R, p_i \in W$ ;  $d_i$  为  $r_i$  的终点位置,  $i \in R, d_i \in W$ ;  $\omega_r$  为任务  $r$  的荷载单元,  $r \in R$ ;  $C_k$  为  $V_k$  的容量;  $t_i^k$  为  $V_k$  执行  $r_i$  的开始时间;  $t_{ij}$  为 RGV 遍历弧  $(i, j)$  的时间;  $t_u$  为装卸单位货物的时间;  $M$  为所有任务完成时间, RGV 最迟完工时间;  $T$  为离散时刻集合,  $T = \{0, 1, \dots, |T|\}$ ,  $|T|$  为  $M$  的上界;

$$y_{ijk}^l = \begin{cases} 1, & V_k \text{ 在任务 } r_l \text{ 和 } r_{l+n} \text{ 之间先取 } r_i, \\ & \text{再取 } r_j \\ 0, & \text{其他} \end{cases}$$

$$z_{akt} = \begin{cases} 1, & V_k \text{ 在 } t \text{ 时刻的位置为 } \lambda \\ 0, & \text{其他} \end{cases}$$

$$x_{ij}^k = \begin{cases} 1, & V_k \text{ 先取 } r_i, \text{再取 } r_j \\ 0, & \text{其他} \end{cases}$$

建立数学模型如下:

$$\min M = \max \left\{ \sum_{(i,j) \in A, k \in K} t_{ij} x_{ij}^k, \sum_{(i,j) \in A, l \in V', k \in K} t_{ij} y_{ijk}^l \right\} \quad (1)$$

s. t.

$$\sum_{i:(i,j) \in A'} y_{ijk}^l = \sum_{i:(i,j+n) \in A'} y_{i,j+n,k}^l \quad (2)$$

$$\forall j \in E^1 / \{l\}, l \in E^1, k \in K$$

$$\forall j \in E^2 / \{l\}, l \in E^2, k \in K$$

$$\sum_{i:(i,j) \in A'} y_{ijk}^l + \sum_{i:(i,j+n) \in A'} y_{i,j+n,k}^l \leq 1 \quad (3)$$

$$\forall j \in E^3 / \{l\}, l \in E^3, k \in K$$

$$\forall j \in E^4 / \{l\}, l \in E^4, k \in K$$

$$y_{ijk}^l = 0 \quad (4)$$

$$\forall (i,j) \in A'$$

$$\forall j \in E^3 / \{l\}, l \in E^3, k \in K$$

$$\forall j \in E^4 / \{l\}, l \in E^4, k \in K$$

$$y_{i,j+n,k}^l = 0 \quad (5)$$

$$\forall (i,j+n) \in A'$$

$$\forall j \in E^4, l \in E^1, k \in K$$

$$\forall j \in E^3, l \in E^2, k \in K$$

$$y_{i,j+n,k}^l = 0 \quad (6)$$

$$\forall (i,j) \in A'$$

$$\forall j \in E^4, l \in E^3, k \in K$$

$$\forall j \in E^3, l \in E^4, k \in K$$

$$\sum_{j:(i,j) \in A} x_{ij}^k = 1, \quad \forall i \in V / \{2n+1\}, k \in K \quad (7)$$

$$\sum_{j:(j,i) \in A} x_{ji}^k = 1, \quad \forall i \in V / \{0\}, k \in K \quad (8)$$

$$\sum_{\lambda \in L} z_{\lambda kt} = 1, \quad k \in K, t \in T \quad (9)$$

$$\sum_{k \in K} z_{\lambda kt} \leq 1, \quad \lambda \in W, t \in T \quad (10)$$

$$z_{0k0} = 1, \quad k = 1 \quad (11)$$

$$z_{|W|+1k0} = 1, \quad k = 2 \quad (12)$$

$$\sum_{\substack{k=\{1\} \\ l \in V / \{2n+1\}}} x_{ij}^k + \sum_{\substack{k=\{2\} \\ l \in V / \{2n+1\}}} x_{li}^k \leq (2 - z_{\lambda kt}) \quad (13)$$

$$\forall i, j \in A, p_i < p_j, \lambda \in W, t \in T$$

$$t_{i+n}^k \geq t_i^k + 2t_u + t_{i,i+n}, \quad \forall i \in P, k \in K \quad (14)$$

$$t_{i \oplus 1}^k \geq t_i^k + t_u, \quad \forall t_{i \oplus 1}^k \neq \emptyset, i \in P, k \in K \quad (15)$$

$$\sum_{r \in R} w_r y_{ijk}^l \leq \sum_{k \in K} C_k x_{ij}^k, \quad (i,j) \in A, k \in K \quad (16)$$

$$y_{ijk}^l \leq x_{ij}^k, \quad (i,j) \in A', l \in P, k \in K \quad (17)$$

$$x_{ij}^k \in \{0,1\}, \quad \forall (i,j) \in A \quad (18)$$

$$y_{ijk}^l \in \{0,1\}, \quad \forall (i,j) \in A', l \in P \quad (19)$$

$$z_{\lambda kt} \in \{0,1\} \quad (20)$$

$$\lambda \in \{0\} \cup W \cup \{2n+1\}, \quad k \in K$$

式(1)为  $V_k$  在消除两类冲突中最后完成所有任务的最少时间;式(2)~(5)为消除 I 类冲突约束,其中,式(2)为后进先出(LIFO)策略,式(3)为先进先出(FIFO)策略,式(4)为交叉任务先入(CFI)策略,式(5)为交叉任务后出(CLO)策略;式(6)为 I 类冲突消除;式(7)和(8)表示每个任务只能被服务

一次;式(9)~(13)为消除 II 类冲突约束,式(9)~(10)为同一时刻一个位置只能被一辆 RGV 访问,式(11)~(12)为初始时刻两辆 RGV 的位置,式(13)表示避免两辆 RGV 碰撞;式(14)表示每个取货任务必须在对应送货任务之前完成;式(15)表明同一货架有多项任务需要装载,则按照 FIFO 原则装载,“ $i \oplus 1$ ”表示紧接着任务  $r_i$  的后一个任务;式(16)为  $V_k$  的容量约束式;式(17)表示局部取送任务和全局取送任务关系;式(18)~(20)为决策变量的取值范围。

### 3 改进型和声搜索算法

和声搜索算法(HS)最早由 Geem 等<sup>[11]</sup>受音乐即兴创作过程的启发而提出,具有原理简单、控制参数少、收敛速度快等优点。然而,该算法仍存在一些缺陷,如搜索后期收敛缓慢、局部搜索能力欠缺、易陷入局部最优,故本文提出改进型和声搜索算法(MHS)求解中大规模问题,以获取较高质量的解。基本步骤如下。

**步骤1** 问题参数初始化。包括和声记忆库大小 HMS、最大迭代次数 NI、和声记忆库取值概率 HMCR、音调微调概率 PAR、微调幅度 BW 以及精度  $\epsilon$ 。

**步骤2** 和声记忆库初始化。结合式(2)~(6),随机生成 HMS 个和声向量,存入 HM。

**步骤3** 划分和声片段。将每个和声向量按照 I 类约束关系,分解为若干和声片段。

**步骤4** 新和声即兴创作。对于每个和声片段,基于 HMCR、PAR 进行即兴创作,形成新的和声。

**步骤5** 更新和声记忆库。将新形成的和声和当前最差和声进行对比,若优于当前最差和声,则替代当前和声。

**步骤6** 判断终止准则。若当前迭代次数  $N$  大于最大迭代次数 NI,则合并所有和声片段形成新的和声记忆库,否则重复步骤 4 和 5。

**步骤7** 变邻域搜索操作。随机选取一个和声向量进行变邻域搜索操作,并更新 HM。

**步骤8** 判断终止准则:若任两个和声对应目标值之差均小于  $\epsilon$ ,则算法终止,否则重复执行步骤 3~7。

#### 3.1 和声编码方案

用  $S_i (i=1,2,\dots,HMS)$  表示和声向量元素,采用 3 层变长编码方式:第 1 层表示图  $G$  顶点层,代表 RGV 访问的线路节点;第 2 层为位置层,代表顶点对应的源点位置  $p$  或终点位置  $d$ ;第 3 层为执行

任务的 RGV 编号层. 以图 1 为例, 若将  $r_1, r_2$  指派给  $V_1$ , 将  $r_3, r_4$  指派给  $V_2$ , 则其中一个可行解可表示为图 2. 其中顶点层的数字 9 为  $2 \times 4 + 1$ , 位置层中间的“11”表示初始时刻,  $V_2$  的位置为  $|W| + 1$ .

顶点层→	0	1	5	2	6	9	0	4	8	3	7	9
位置层→	0	1	3	2	4	0	11	9	4	5	9	11
RGV 编号层→	1	1	1	1	1	1	2	2	2	2	2	2

图 2 和声编码图解

Fig. 2 Encoding mode for harmony

给定任务  $n$ , 根据有向图  $G$  的定义以及两辆 RGV 的起讫点位置, 可得和声的最大长度  $L_{\max} = 2n + 4$ . 当  $n = 1$  时, 可知  $L_{\min} = 5$ , 即一辆 RGV 的初始点和执行一个完整任务的源点和终点. 因此和声  $S_i$  的长度范围为  $[L_{\min}, L_{\max}]$ .

编码方案对应的解如图 3 所示, 横线部分表示 RGV 进行装或卸操作. 初始时刻,  $V_1$  从 0 位置出发, 完成搬运任务后返回至原位置;  $V_2$  从  $|W| + 1$  出发, 完成分配的任务后返回至出发点.  $V_1$  对应的路径为  $0 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 3 \rightarrow 0$ , 执行的任务顺序为 ②→①; RGV<sub>2</sub> 对应的路径为  $11 \rightarrow 9 \rightarrow 4 \rightarrow 5 \rightarrow 9 \rightarrow 11$ , 执行的任务顺序为 ④→③.

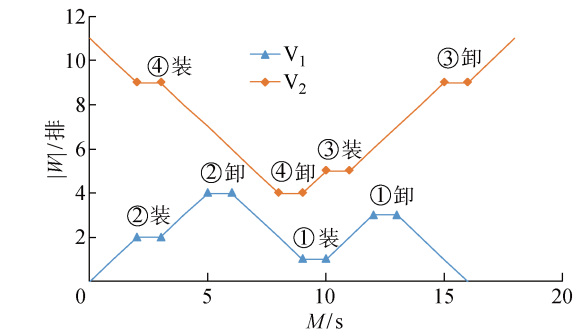


图 3 编码方案对应的解

Fig. 3 Corresponding solution of coding mode

3.2 和声记忆库初始化

为生产初始可行解, 本文采用贪婪随机自适应搜索算法 (GRASP) 初始化和声记忆库. 具体步骤如下.

**步骤 1** 随机生成任务矩阵  $R^n = [r_1 \ r_2 \ \cdots \ r_n]^T$ ,  $r_n = (\alpha_n, P_n, \beta_n, D_n)$ ,  $\alpha_n = \text{rand}(1, 2)$ ,  $P_n = \text{rand}(1, |W|)$ ,  $\beta_n = \text{rand}(1, 2)$ ,  $D_n = \text{rand}(1, |W|)$ .

**步骤 2** 令  $R_s^n \leftarrow \text{sort}(P)$ ,  $\tau = \text{floor}(n/2)$ , 若  $P_n \leq \tau$ ,  $R_1^{n_1} \leftarrow R_s^n$ , 否则,  $R_2^{n_2} \leftarrow R_s^n$ . 其中,  $n_1 + n_2 = n$ .

**步骤 3** 令  $h_1 \leftarrow \text{rand}(1, \text{size}(R_1^{n_1}, 1)R_1^{n_1})$ .

**步骤 4** for  $i = 2$  to  $\text{size}(R_1^{n_1}, 1)$

(1) 根据定理 1 及式 (2)~(6), 从  $R_1^{n_1}$  中筛选出与  $h_1$  中最新添加的任务不存在 I 类冲突的任务集  $R_{\text{tmp}}^{n_1}$ .

(2) 若  $R_{\text{tmp}}^{n_1} = \emptyset$ , 则将  $d_{i-1}$  设置为新源点, 选取  $r_i \in R_1^{n_1}$  且满足两点距离为  $\min\{p_i - d_{i-1}\}$ , 添加到  $h_1$ , 转至 (1), 否则转至 (3).

(3) 若  $h_1(p) < h_1(d)$  且  $h_1(p) \leq p_i < h_1(d)$  且  $p_i < d_i$ , 则将  $\lambda_i = \min\{p_i - h_1(p)\}$  添加到  $h_1$ , 并在  $R_1^{n_1}$  中删除  $r_i$ .

(4) 根据定理 1 及式 (2)~(6) 判定终点, 若终点为  $d_{i-1}$ , 则将  $d_{i-1}$  设置为新源点,  $d_i$  为新终点, 返回 (1). 若终点为  $d_i$ , 则设  $d_i$  为新源点,  $d_{i-1}$  为新终点, 转至 (1).

(5) 若  $R_1^{n_1} = \emptyset$ , 转至步骤 5, 否则转至 (1).

**步骤 5** 计算路径  $h_1$  和  $h_2$  的时空坐标  $p_{1,i}^m$ , 其预计开始时间  $T_{1,i}^{m,\text{st}} = |p_{1,i}^m - p_{1,i}^{m-1}|/v$ , 结束时间  $T_{1,i}^{m,\text{et}} = T_{1,i}^{m,\text{st}} + t_u$  ( $v$  为 RGV 的速度,  $t_u$  为货物的装卸时间); 路径距离  $\text{dis}_{1,i} = \sum_{m=1}^{2n_1} |p_{1,i}^m - p_{1,i}^{m-1}|$ , 其中  $p_{k,i}^0 = 0$ , 路径的总时间  $T(R^1) = \text{dis}_{1,i}/v$ .  $h_{2,j}$  的时间参数和距离参数计算与  $h_{1,i}$  相同.

**步骤 6** 将  $h_1$  和  $h_2$  转化成在  $[L_{\min}, L_{\max}]$  上的编码, 一并存入 HM, 实现和声记忆库的初始化.

3.3 即兴创作产生新和声

对基本 HS 进行新和声创作, 需在定义域随机生成一个新的和声  $S^{\text{new}}$ , 由于编码的特殊性及双冲突的存在, 若采用基本 HS 算法进行即兴创作产生新和声, 不仅会降低解的质量, 也会降低收敛速度. 为此, 本文借鉴文献 [12] 的策略, 将 HM 按和声长度  $L_i$  划分成若干个大小相异的和声片段 Sub\_HM. 为了防止算法过早收敛、陷入局部最优, 针对每个 Sub\_HM, 结合参数 HMCR、PAR 等进行新和声创作, 并更新 Sub\_HM.

针对每个 Sub\_HM, 新和声  $S^{\text{new}}$  以 HMCR 概率继承现有 Sub\_HM 中和声, 并以 PAR 概率进行微调. 同时, 对  $S^{\text{new}}$  进行邻域搜索, 新和声  $S^{\text{new}}$  以  $(1 - \text{HMCR})$  的概率应用 GRASP 算法进行创作. 具体流程如下.

$$r_1, r_2, r_3 = \text{rand}()$$
$$\text{for } j = 2 \text{ to } 2n + 2 \text{ do}$$
$$\text{if } r_1 < \text{HMCR then}$$
$$S^{\text{new}} \leftarrow S^a \text{ where } a \in (1, 2, \dots, \text{HMS})$$
$$\text{if } r_2 < \text{PAR then}$$
$$S^{\text{new}} \leftarrow S^{\text{new}} \pm r_3 BW$$

```

end if
else
 $S^{\text{new}} \leftarrow \text{Sub\_HM} + \text{GRASP}$ 
if  $f(S^{\text{new}}) < f(S^{\text{worst}})$ 
 $S^{\text{worst}} \leftarrow S^{\text{new}}$  // * 更新 HM
end if
end for

```

### 3.4 变邻域搜索操作

为了进一步提升解的质量和保持解的多样性,引入变邻域搜索(VNS)机制更新和声记忆库,包括 4 种移动操作和 1 个变异操作。为方便描述,定义  $(r_i^+, r_i^-) \in S_i$  (图中用  $i^+, i^-$  表示) 为一个任务对,  $(r_i^+, r_k^+, \dots, r_i^-, r_k^-) \in S_i$  为一个组合任务。

(1) 重置任务对。随机选取一个任务对,插入到其他合适的位置,如图 4(b) 所示,将初始解(图 4(a))中  $(r_3^+, r_3^-)$  插入到  $r_2^-$  之后的位置。

(2) 重置组合任务。随机选取一个完整组合,插入到线路其他位置,如图 4(c) 所示,将初始解(图 4(a))中  $(r_1^+, \dots, r_2^-)$  插入到  $r_3^-$  之后的位置。

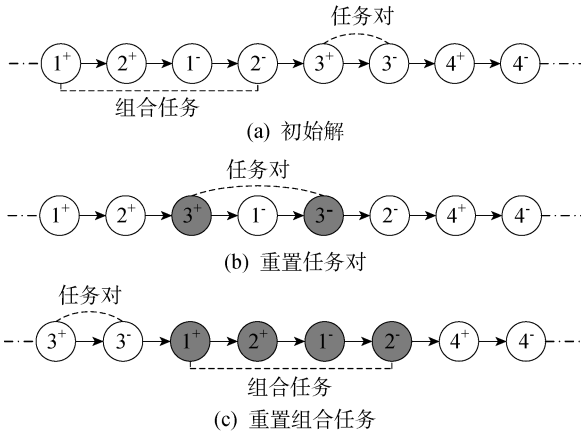


图 4 任务对和组合任务的重置操作

Fig. 4 Re-located operations of couple and component

(3) 任务对交换。随机选取两个任务对,交换对应的位置,如图 5(b) 所示,交换了初始解(图 5(a))中  $r_1$  和  $r_3$  的位置。

(4) 组合任务交换。随机选取两个组合任务,并交换对应的位置,如图 5(c) 所示,交换了初始解(如图 5(a))中  $C_x$  和  $C_y$  的位置。

(5) 变异操作。设原变量  $S^{\text{old}}$  经移动操作后得到  $S^{\text{mv}}$ , 若  $f(S^{\text{mv}}) < f(S^{\text{old}})$ , 则  $S^{\text{old}} = S^{\text{mv}}$ , 否则  $S^{\text{mv}}$  作为候选变异变量。引入变异概率  $P_m$ , 若  $\text{rand}(0, 1) < P_m$ , 则随机选取一任务对进行变异,以改变其小车分配,添加到 HM。变异过程如图 6 所示。

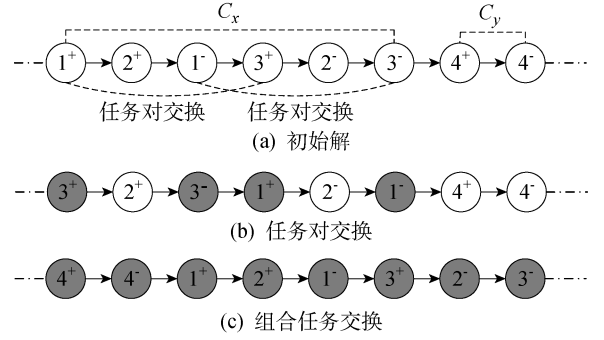


图 5 任务对和组合任务的交换操作

Fig. 5 Swap operations of couple and component

变异前						变异后					
0	1	5	2	6	9	0	1	5	2	6	9
0	1	3	2	4	0	0	1	3	2	4	0
1	1	1	1	1	1	1	2	2	1	1	1

图 6 和声向量变异操作

Fig. 6 Mutation operation of harmony vector

重置任务对(ope1)、重置组合任务(ope2)、交换任务对(ope3)和交换组合任务(ope4)4 个操作的执行控制为

$$S_i \leftarrow \begin{cases} \text{ope1}, & 0 \leq q < \frac{1}{3} \text{PAR} \\ \text{ope2}, & \frac{1}{3} \text{PAR} \leq q < \frac{2}{3} \text{PAR} \\ \text{ope3}, & \frac{2}{3} \text{PAR} \leq q < \text{PAR} \\ \text{ope4}, & \text{PAR} \leq q \leq 1 \end{cases} \quad (21)$$

式中:  $q$  为  $[0, 1]$  之间的随机数。

经以上操作得到的新向量有可能不可行,为保证解的多样性,允许一定数量不可行解的存在。为此,本文引入罚函数法,通过惩罚函数,所有的不可行解向量都倾向于朝着可行解的方向搜索,从而完成不可行解向可行解的转化。

令 I 类约束式(2)~(6)分别用  $g_1, g_2, \dots, g_5$  表示, II 类约束式(13)用  $u$  表示。则罚函数可构建为  $\max \tilde{M} =$

$$-M + \gamma \left( \sum_{i=1}^5 \max\{0, g_i\} + \max\{0, u\} \right) \quad (22)$$

式中:  $\gamma > 0$  为违反 I 类约束和 II 类约束的惩罚系数。

### 3.5 MHS 的核心流程

基于以上分析,给出 MHS 的核心流程。

参数初始化: HMS, HMCR, PAR, NI,  $S^{\text{best}}$



按照 GRASP 算法,初始化和声记忆库  
按  $r_n$  的数量,规划 Sub\_HM  
for  $i=1$  to HMS do  
    if  $f(S_i)<f(S^{\text{best}})$  then  
         $S^{\text{best}}\leftarrow S_i$   
    end if  
end for  
新和声创作  $S^{\text{new}}$  (见 3.3)  
更新和声记忆库(见 3.4)  
if  $S^{\text{new}}$  不可行) then  
    计算  $\widetilde{M}$   
end if  
if  $f(S^{\text{new}})<f(S^{\text{best}})$  then  
     $S^{\text{best}}\leftarrow S^{\text{new}}$   
end if  
until 达到终止标准  
return  $S^{\text{best}}$

## 4 仿真实验及分析

为评价本文提出的 MHS 算法的有效性,结合文献[10]生成如下参数: $V_k(k=\{1,2\})$ 速度  $v=1$  m/s;最大容量  $C_k=2$ ;装卸时间  $t_u=1$  s;货架间距为 0.1 m,货架宽为 1 m. 分别取不同的  $n$  和  $|W|$  的组合,做仿真实验,并进行统计分析.

算法采用 MATLAB(R2014a)编程实现,仿真环境是主频为 2.60 GHz,内存为 16 GB 的便携式计算机.

### 4.1 算法有效性验证

为验证 MHS 的有效性,将问题规模  $|W|$  设定为 5 和 10; $n$  设为 8、10、12、14、16,且均匀分布于  $W$ ,进行小规模问题求解. 将运行结果与 CPLEX12.6 进行比较,针对每个  $|W|$  和  $n$  的组合,CPLEX 的最大求解时限设为 2 h,MHS 算法运行 10 次取其平均值,统计结果如表 1 所示. 其中,最优解是 CPLEX 求解所得,可以看出,MHS 运行得出的结果与最优解的最大偏差为 4.68%,小于 5%,验证了算法的有效性. 另外,“\*”表示 CPLEX 无法在规定的时限内获得最优解,只能获得最优解的下界,因此,有必要构建智能优化算法求解中、大规模的满意解.

令问题规模  $|W|\in\{15,20\}$ , $n\in\{20,50,80,110\}$ 进行中、大规模问题的仿真,以验证 MHS 的有效性. 由于本文研究的问题没有标准的算例库,考虑到本调度问题与同时取送 (SPD) 问题有一定的相似

表 1 小规模问题实验结果

Tab. 1 Results of small scale instances

$ W $	$n$	CPLEX	MHS	偏差/%
5	8	42.5	46.6	4.25
5	10	55.3	58.9	3.83
5	12	76.8	80.2	3.56
5	14	79.1	82.4	3.47
5	16	91.3	92.7	1.52
10	8	24.3	28.6	4.35
10	10	31.2	35.1	3.97
10	12	34.4	37.9	3.63
10	14	38.6 *	42.4	4.68
10	16	45.7 *	49.2	4.36

性,而文献[13]的混合遗传算法 (HGA) 和文献[14]的粒子群算法 (PSO) 在求解取送问题 (PDP) 上具有较好性能,故将其用于本文的中、大规模问题求解. 为了让文献的算法更好地求解该问题,初始种群同样由 GRASP 算法产生,编码方案做了细微的修改,并加入了 3.4 节的变邻域搜索操作,以彰显与目标算法 (MHS) 的可比性. 表 2 所示为将 MHS 获得的结果和 HGA、PSO 得到的结果进行了对比统计. 从表中可以看出:随着问题的扩大,MHS 的寻优能力明显比 HGA 和 PSO 突出;当  $n=110$  时,最大相对偏差达到 11.02% 和 9.92%,一定程度上验证了 MHS 求解中、大规模问题的有效性.

表 2 MHS、HGA 和 PSO 的实验结果统计

Tab. 2 Experimental results of MHS, HGA and PSO

$ W $	$n$	MHS	HGA		PSO	
			M/s	偏差/%	M/s	偏差/%
15	20	121.4	123.1	1.46	122.8	1.18
15	50	313.6	325.1	3.85	327.6	4.52
15	80	453.2	475.8	5.01	482.7	6.56
15	110	752.6	823.4	9.52	816.3	8.85
20	20	125.8	128.9	2.63	130.1	3.44
20	50	321.5	344.7	7.31	341.9	6.43
20	80	465.3	508.6	9.42	502.8	8.12
20	110	942.6	1045.4	11.02	1 035.2	9.92

### 4.2 MHS 收敛性验证

为进一步验证 MHS 的收敛性能,令问题规模  $|W|=15,n=50$  以及  $|W|=20,n=110$ ,绘制两种组合的收敛曲线,将其与 HS 常见的两种改进型算法,即改进型和声搜索算法 (IHS) 和全局最优和声搜索算法 (GHS) 进行对比,结果如图 7 所示.

由图 7 可见,3 条曲线的收敛趋势大抵一致,主要原因在于都是在基本 HS 的基础上对其后期收敛速度慢进行改进.中小规模下,3 种算法都能在较短时间内搜索到最优解,随着规模的增大,本文融入的变邻域搜索机制使得 MHS 能够在搜索深度上更胜一筹,如图 7(b)所示.实验结果表明,MHS 具备良好的收敛性能.

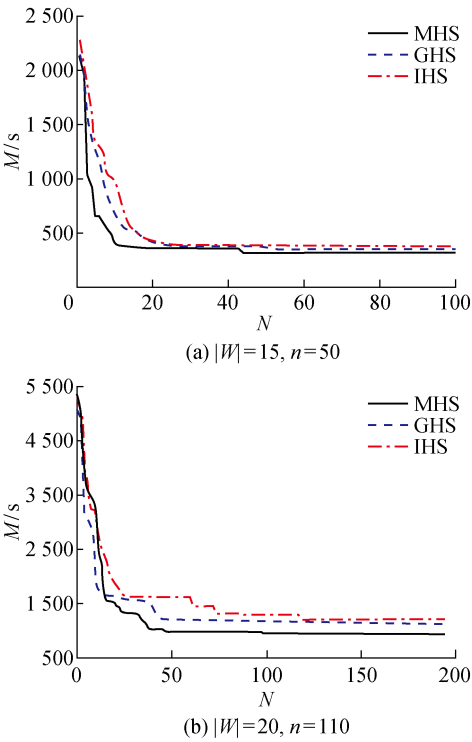


图 7 算法的收敛性能

Fig. 7 Convergence performance for algorithms

4.3 系统搬运效率分析

为评估本文调度方法对系统效率的影响情况,引入 AHT 和 ART 两个参数作为衡量基准. AHT 表示任务的平均搬运时间,其值越小,表示系统的运行效率越高;ART 表示为了避免 II 类冲突,RGV 之间相互避让,导致货物在车上的滞留时间,其值越小,说明调度越有效.

令  $|W|$  分别为 6,8,10,12,14,16,18,20,任务  $n=50$ ,仿真实验结果如图 8 所示,其中,  $t'$  为任务所需平均时间.

由图 8 可见:当  $|W|$  从 6 增加到 16 时,AHT 呈逐渐下降的趋势,主要原因可能是空间狭窄,任务都集中在较少的货架上,两辆 RGV 相互干扰,大部分的时间花费在等待上;而  $|W|>16$  时,AHT 开始逐渐增加,空间问题逐步得到缓解,两辆 RGV 基本可以按照各自的预定线路进行搬运;ART 则是逐渐下降,当  $|W|$  从 6 增加到 16 时下降缓慢;当

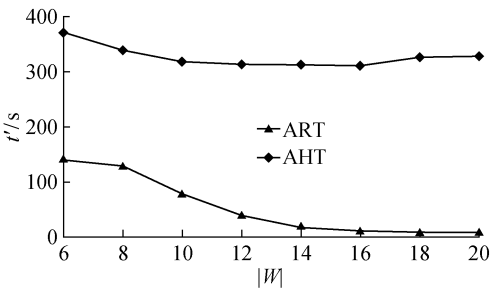


图 8 AHT 和 ART 与  $|W|$  的关系

Fig. 8 Relationship among AHT, ART and  $|W|$

$|W|>16$  时,其值下降趋势明显;同 AHT 变化的原因一致,RGV 之间的干扰降低,货物的滞留时间也随之减少,货物在小车上的滞留时间几乎等于货物的搬运时间.从上述结果来看,两辆 RGV 同时作业,货架数量  $|W|>16$  比较合适.当货架数量比较少(少于 16)时,不太适合采用多 RGV 同时作业.

5 结语

本文研究了自动存取系统多载量 RGV 避碰调度问题,在对问题描述的基础上建立了混合整数规划模型.在算法部分,提出改进型和声搜索算法,在局部搜索部分融入了变邻域搜索机制,以提高基本和声搜索算法后期的收敛速度和解的质量.同时,引入 AHT 和 ART 两个参数作为系统搬运效率的评价指标,当货架数量大于 16 时平均搬运时间呈缓慢递增趋势,货物在 RGV 内的平均滞留时间呈持续下降趋势,说明了调度的有效性.

参考文献:

[1] ROODBERGEN K J, VIS I F A. A survey of literature on automated storage and retrieval systems[J]. *European Journal of Operational Research*, 2009, 194 (2): 343-362.

[2] 郑向东, 谢飞. 均料分配系统中环形同轨自动导引车的应用研究[J]. *物流技术*, 2011, 30(7): 187-189. ZHENG Xiangdong, XIE Fei. Application of circular co-rail guided vehicle in load-sharing systems[J]. *Logistics Technology*, 2011, 30(7): 187-189.

[3] LEE S G, DE SOUZA R, ONG E K. Simulation modelling of a narrow aisle automated storage and retrieval system (AS/RS) serviced by rail-guided vehicles[J]. *Computers in Industry*, 1996, 30 (3): 241-253.

[4] LEE J. Dispatching rail-guided vehicles and scheduling jobs in a flexible manufacturing system [J]. *International Journal of Production Research*, 1999, 37(1): 111-123.



- [5] CHEN F F, HUANG J, CENTENO M A. Intelligent scheduling and control of rail-guided vehicles and load/unload operations in a flexible manufacturing system[J]. **Journal of Intelligent Manufacturing**, 1999, 10(5): 405-421.
- [6] DOTOLI M, FANTI M P. A coloured Petri net model for automated storage and retrieval systems serviced by rail-guided vehicles: A control perspective[J]. **International Journal of Computer Integrated Manufacturing**, 2005, 18(2/3): 122-136.
- [7] LIU Y K, LI S S, LI J, *et al.* Operation policy research of double rail-guided vehicle based on simulation[C]// **2010 International Conference on E-Product E-Service and E-Entertainment**. Henan, China: IEEE, 2010.
- [8] KUNG Y, KOBAYASHI Y, HIGASHI T, *et al.* Order scheduling of multiple stacker cranes on common rails in an automated storage/retrieval system[J]. **International Journal of Production Research**, 2014, 52(4): 1171-1187.
- [9] GAO Q, LU X W. The complexity and on-line algorithm for automated storage and retrieval system with stacker cranes on one rail[J]. **Journal of Systems Science and Complexity**, 2016, 29(5): 1302-1319.
- [10] HU W H, MAO J F, WEI K J. Energy-efficient rail guided vehicle routing for two-sided loading/unloading automated freight handling system[J]. **European Journal of Operational Research**, 2017, 258(3): 943-957.
- [11] GEEM Z W, KIM J H, LOGANATHAN G V. A new heuristic optimization algorithm: Harmony search[J]. **Simulation**, 2001, 76(2): 60-68.
- [12] 周炳海, 徐佳惠, 彭涛. 基于新型线边集成超市的周期性物料配送优化[J]. **吉林大学学报(工学版)**, 2018, 48(2): 588-595.
- ZHOU Binghai, XU Jiahui, PENG Tao. Optimization of cyclic part feeding with novel line-integrated supermarket[J]. **Journal of Jilin University (Engineering and Technology Edition)**, 2018, 48(2): 588-595.
- [13] KARAOGLAN I, ALTIPARMAK F. A hybrid genetic algorithm for the location-routing problem with simultaneous pickup and delivery[J]. **Industrial Engineering and Management Systems**, 2011, 10(1): 24-33.
- [14] AI T J, KACHITVICHYANUKUL V. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery[J]. **Computers & Operations Research**, 2009, 36(5): 1693-1702.

(本文编辑:黄伟)