

文章编号:1007-5429(2014)06-0070-08

基于分区法的 2-RGV 调度问题的模型和算法

陈 华^{1,2,3}

(1. 西安交通大学 管理学院, 陕西 西安 710049; 2. 机械制造系统工程国家重点实验室, 陕西 西安 710049;
3. 过程控制与效率工程教育部重点实验室, 陕西 西安 710049)

摘要: 以某救灾物资自动化立体仓库为背景, 研究了出库过程中的 2-RGV 调度问题。为避免运行于直线轨道上的两辆穿梭车发生碰撞, 将存取系统划分为不重叠的两个区域, 并将两个区域内的运送分别分配给两辆穿梭车。建立了基于区域划分的 2-RGV 调度问题的混合整数线性规划模型, 目标是 최소화所有物料的总出库时间。提出了一个混合遗传算法求解该问题, 并采用包含穿梭车顺序、穿梭车分配及出库站分配的染色体编码方式。算例试验显示提出的模型和算法能够有效求解该问题。

关键词: 穿梭车调度; 分区法; 混合整数线性规划模型; 混合遗传算法; 自动化立体仓库

中图分类号: TP278; O221.1

文献标识码: A

Model and Algorithm for Two Rail-guided Vehicles Scheduling Problem Based on a Partitioning Approach

CHEN Hua^{1,2,3}

(1. School of Management, Xi'an Jiaotong University, Xi'an 710049, China; 2. The State Key Laboratory for Manufacturing Systems Engineering, Xi'an 710049, China; 3. The Key Lab of the Ministry of Education for Process Control & Efficiency Engineering, Xi'an 710049, China)

Abstract: This paper studies the two rail-guided vehicles (RGV) scheduling problem during the retrieval operations in a disaster relief automated storage and relief system. To avoid collisions between the two RGVs running on the same rail, the storage and retrieval system is divided into two non-overlapping zones, and the RGV moves of the two zones are dispatched to the two RGVs separately. The 2-RGV scheduling problem based on partitioning approach is formulated into a mixed integer linear programming model. The objective is to minimize the total retrieval time for all the items to be retrieved. To solve the problem, a hybrid generic algorithm is proposed and a chromosome representation, which contained the information of the RGV move sequence, the RGV assignments and the output station assignments, is used. Computational experiments indicate that the model is effective and the hybrid generic algorithm performs well on randomly generated problem instances.

Key words: rail-guided vehicle scheduling; partitioning approach; mixed integer linear programming model; hybrid generic algorithm; automated storage and retrieval system

收稿日期: 2014-03-20; 修回日期: 2014-06-20

基金项目: 国家自然科学基金重大项目 (71390333)

作者简介: 陈华 (1983-), 女, 陕西人, 博士研究生, 主要研究方向为物流与生产运作管理。

1 引言

穿梭车(RGV)由于其价格低廉、灵活和方便的特点,被广泛应用于自动化立体仓库(AS/RS)中^[1]。RGV系统是AS/RS物料搬运系统的重要组成部分,其作业效率对AS/RS的吞吐率具有重要的影响作用。随着信息技术和自动化技术的发展,单辆RGV可以高速准确地运行,然而,当系统中存在多辆RGV时,RGV之间极易产生相互碰撞或死锁,因此,RGV系统往往成为整个仓储系统的瓶颈。有效调度RGV能够避免碰撞和提高仓库性能。

本文的研究背景为某救灾物资立体仓库,该仓

库的RGV系统由直线轨道上往复行驶的两辆RGV构成。图1给出了该AS/RS的布局图。由于在这类仓库中,救灾物资的入库、补给和替换任务较零散,单次作业的入、出库物资量较小,而一旦有灾害发生就需要在较短时间出库大量物资。因此,这类仓库更关注如何提高出库作业的效率。本文只考虑AS/RS出库过程中的RGV调度问题,存储作业不作考虑。本文将直线轨道上往复运行的两辆RGV的调度问题记为2-RGV调度问题。由于RGV不能相互越过,也不能于同一时间停留在同一位置,因此2-RGV调度的重点是碰撞避免。

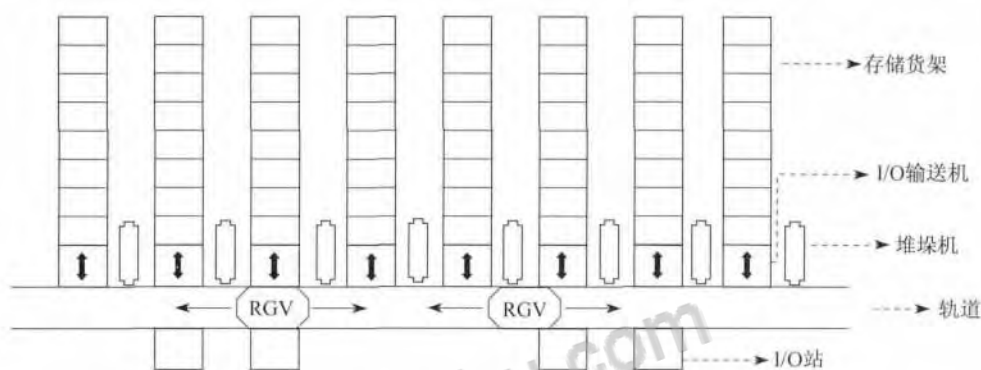


图1 救灾物资立体仓库布局图

截止目前关于RGV的研究主要集中于RGV的设计、控制、派遣规则及路径选择等方面,RGV调度问题的研究文献并不多见。Lee SG建立了自动化立体仓库的仿真模型以确定最优RGV数量和最大化系统吞吐率的最优策略^[2]。J Lee和R Maneesavet提出制造系统背景下的5种RGV分派规则,并通过仿真实验评价了这5种规则与3种普通调度规则^[3]。Dotoli M和Fanti MP应用着色赋时Petri网提出了RGV与堆垛机的物料搬运系统的建模框架,使用了几种控制策略来改进系统效率和避免环轨RGV的碰撞和死锁^[4]。Chen FF等提出了一个制造系统背景下的RGV调度和控制系统^[5]。Dotoli M和Fanti MP研究了RGV系统的死锁避免策略和死锁保护策略^[6]。Liu YK等研究了2-RGV系统的作业策略:路径分区模式和路径整合模式,并通过仿真实验对比了两种策略的优劣^[7]。吴长庆等提出了基于双层着色赋时petri网的RGVS系统的动态模型^[8]。杨少华等人应用排队论对环轨多车情况下的穿梭车数量和能力进行了分析^[9]。吴焱明等研究了环轨RGV调度问题^[10]。前述文献局限于利用车辆派遣规则、路径选择规则以

及智能优化算法进行仿真模拟,这些方法的优点是可以得到一个可行方案,缺点是不能对问题进行精确求解。

类似的碰撞避免问题也出现在双抓钩周期调度问题中,当两个抓钩在一条化工处理线上搬运工件时,极有可能发生相互碰撞,进而需要对抓钩的运送进行冲突避免的调度以达到周期最短的目的。Lei L和Wang在求解双抓钩周期调度问题时提出分区的方法,将生产线分为两个区域,并将两个区域内的运送分别分配给两个抓钩,找出两个区域抓钩调度问题的公共周期,通过比较不同的分区结构找到最好解^[11]。Zhou ZL使用类似的分区方法对双抓钩周期调度问题进行精确求解^[12,13]。

借鉴类似方法,本文使用分区的方法求解2-RGV的调度问题。求解的基本思路是将存取系统(所有货架和出库站)按照轨道上某点一分为二,将分界点左右两侧的运送分别分配给两辆RGV,此时两个区域内的RGV的运送路线不会产生重叠,只有当分界点恰好位于某入/出库(I/O)输送机或某个I/O站位置时才有可能发生碰撞,只要保证两辆RGV不在同一时间访问同一位置既可避免碰撞。

图 2 给出了一种分区的示意图。分区法能够有效降低 RGV 相互碰撞的概率以及求解问题的复杂度。针对分区情况下的 RGV 运送特征, 本文建立了基

于分区法的 2-RGV 调度问题的混合整数线性规划 (MILP) 模型, 目标是最小化物料的总出库时间。

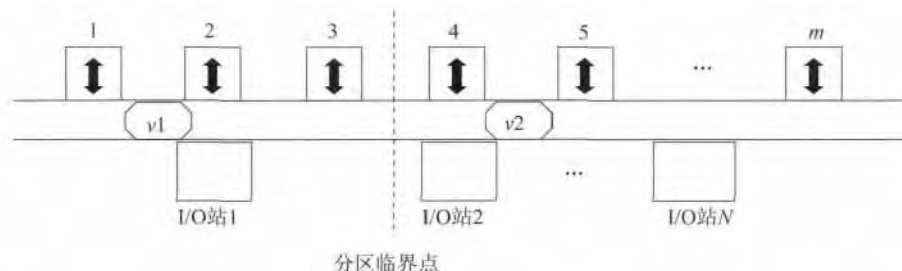


图 2 分区的 RGV 分配示意图

2 问题描述

本文研究的问题可以描述如下: 该救灾物资立体仓库有 N 个 I/O 站, m 条巷道 (详见图 1)。每条巷道内有一排单层货架, 并配备一台堆垛机。巷道 a 内的货架和 I/O 输送机分别记为货架 a 和 I/O 输送机 a 。两辆 RGV 负责物料的运输作业, 记左边的 RGV 为 $v1$, 右边的为 $v2$ 。 J_{aj} 表示存储在货架 a 上货格 j 内的物料。分区情况下, 出库时, 物料先由堆垛机拣取至 I/O 输送机上, 后由区域内空闲的 RGV 运输至该区域内的某个 I/O 站。若当时 RGV 忙, 物料会暂存于 I/O 输送机上。类似地, 若 RGV 取货时堆垛机尚未完成拣货, 则 RGV 需要等待。因此, 为了减少 RGV 潜在的等待时间及不必要行驶时间, RGV 调度时除了要考虑碰撞避免还需考虑 RGV 与堆垛机的协同配合。定义一个 RGV 运送为 RGV 的一系列动作: 从 I/O 输送机处装载物料、运输物料至 I/O 站及卸载物料。未搬运物料的 RGV 运送称为空运送。运送 aj 代表将物料 J_{aj} 从 I/O 输送机 a 运送到某个 I/O 站的一个 RGV 运送。RGV 调度的目的是确定运送顺序、运送开始时间, 及每台堆垛机的取货顺序。

2.1 基本假设

调度问题必须满足以下假设:

- (1) 每个货格最多只能容纳一单位物料。
- (2) 堆垛机、RGV、I/O 输送机的容量均为一单位。
- (3) 每个 I/O 输送机和 I/O 站同一时间只能接待一辆 RGV。
- (4) 假设 RGV 匀速行驶, 且 RGV 在执行载货运送和空运送时的速度是恒定的。
- (5) 一旦 RGV 开始一个运送, 运送完成前不能

中途停止。

(6) 堆垛机从相邻两个货格里拣取物料所需的时间间隔为定值。

(7) 0 时刻 RGV 可以在轨道的任意位置, 所有 I/O 输送机上没有货物且所有堆垛机可以开始拣货。

2.2 参数和变量

模型中的变量和参数的定义如下:

NUM 待出库物料总数。

m 货架数。

N I/O 站个数。

n_a 货架 a 上出库物料数量。

a 货架编号, $a=1, \dots, m$ 。

e I/O 站编号, $e=1, \dots, N$ 。

v RGV 编号, $v=1, 2$ 。

j 货格编号, $j=1, \dots, n_a$ 。

V RGV 的平均速度。

P_a I/O 输送机 a 的位置。

q_e I/O 站 e 的位置。

M 一个很大的正整数。

t_{ae} RGV 从 I/O 输送机 a 行驶到 I/O 站 e , 或从 I/O 站 e 行驶到 I/O 输送机 a 所需要的时间。

u RGV 装载或卸载物料所需的平均时间。

r_{aj} 堆垛机拣取物料 J_{aj} 所需要的时间。

θ 两辆 RGV 之间的安全距离。

决策变量:

P_{aj}^s 运送 aj 的开始位置。

P_{aj}^c 运送 aj 的完成位置。

R_{aj} 运送 aj 的开始时间。

C_{aj} 运送 aj 的完成时间。

C_{max} 所有运送中最晚完成的运送的完成时间。

$$x_{aji} = \begin{cases} 1 & \text{货架 } a \text{ 上第 } i \text{ 次出库的物料是 } J_{aj} \\ 0 & \text{其他} \end{cases}$$

$$w_{aje} = \begin{cases} 1 & \text{物料 } J_{aj} \text{ 被运输至 I/O 站 } e \\ 0 & \text{其他} \end{cases}$$

$$z_{ajv} = \begin{cases} 1 & \text{运送 } aj \text{ 由 RGV } v \text{ 来执行} \\ 0 & \text{其他} \end{cases}$$

本文引入以下变量,用来避免 RGV 碰撞,这些变量定义的条件为 $a, b = 1, \dots, m, j = 1, \dots, n_a, k = 1, \dots, n_b, a \neq b$, 或 $\forall a = b, j \neq k$ 。

$$y_{aj, bk}^s = \begin{cases} 1 & \text{若运送 } aj \text{ 早于运送 } bk \text{ 开始} \\ 0 & \text{其他} \end{cases}$$

$$y_{aj, bk}^s$$

=

$$\begin{cases} 1 & \text{若运送 } aj \text{ 的开始时间早于运送 } bk \text{ 的完成时间} \\ 0 & \text{其他} \end{cases}$$

$$y_{aj, bk}^s$$

=

$$\begin{cases} 1 & \text{若运送 } aj \text{ 的完成时间早于运送 } bk \text{ 的开始时间} \\ 0 & \text{其他} \end{cases}$$

$$y_{aj, bk}^c = \begin{cases} 1 & \text{若运送 } aj \text{ 早于运送 } bk \text{ 完成} \\ 0 & \text{其他} \end{cases}$$

2.2.1 约束条件

(1) 运送顺序约束

$$R_{bk} - R_{aj} \leq M y_{aj, bk}^s \quad (1)$$

$$y_{aj, bk}^s + y_{bk, aj}^s = 1 \quad (2)$$

$$C_{bk} \geq C_{aj} + M(y_{aj, bk}^c - 1) \quad (3)$$

$$y_{aj, bk}^c + y_{bk, aj}^c = 1 \quad (4)$$

$$C_{bk} \geq R_{aj} + M(y_{aj, bk}^s - 1) \quad (5)$$

$$R_{bk} \geq C_{aj} + M(y_{aj, bk}^c - 1) \quad (6)$$

$$y_{aj, bk}^s + y_{bk, aj}^s = 1 \quad (7)$$

约束(1)至(7)保证变量的 $y_{aj, bk}^s, y_{aj, bk}^c, y_{aj, bk}^s$ 的正确地定义。

(2) 堆垛机作业约束

$$\sum_{i=1}^{n_a} x_{aji} = 1 \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (8)$$

$$\sum_{j=1}^{n_a} x_{aji} = 1 \quad a = 1, \dots, m; i = 1, \dots, n_a \quad (9)$$

$$R_{aj} \geq r_{aj} + M(x_{aji} - 1) \\ a = 1, \dots, m; j = 1, \dots, n_a \quad (10)$$

$$R_{aj} \geq r_{aj} + R_{ah} + M(x_{aji} + x_{ah, i-1} - 2) \\ a = 1, \dots, m; i, j, h = 1, \dots, n_a; h \neq j; i \neq 1 \quad (11)$$

约束(8)和约束(9)保证堆垛机每次只拣取一件物料,且每件物料只能被拣取一次。约束(10)说明

当 J_{aj} 是货架 a 上第一个出库的物料时,运送 aj 的开始时间至少为 r_{aj} 。当 J_{ah} 和 J_{aj} 是堆垛机从货架 a 上连续两次拣取的物料,且 J_{ah} 先被拣取时,约束(11)保证在运送 ah 开始后和运送 aj 开始前,堆垛机有足够的时间拣取 J_{aj} 。

(3) 单辆 RGV 的运送约束

$$\sum_{v=1}^2 z_{ajv} = 1 \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (12)$$

$$\sum_{e=1}^N w_{aje} = 1 \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (13)$$

$$C_{aj} = R_{aj} + \sum_{e=1}^N t_{ae} w_{aje} + 2u \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (14)$$

$$R_{bk} \geq R_{aj} + \sum_{e=1}^N t_{ae} w_{aje} + 2u +$$

$$\sum_{e=1}^N t_{be} w_{aje} + M(z_{ajv} + z_{bkv} + y_{aj, bk}^s - 3)$$

$$a, b = 1, \dots, m; j = 1, \dots, n_a; k = 1, \dots, n_b; v = 1, 2; a \neq b, \text{ 或 } \forall a = b, j \neq k \quad (15)$$

$$C_{aj} \leq C_{\max} \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (16)$$

变量 z_{ajv} 给出 RGV 的分配。约束(12)保证每件物料只能分配给一辆 RGV。变量 w_{aje} 给出 I/O 站的分配。约束(13)保证每件物料只能被搬运至一个 I/O 站。约束(14)给出运送完成时间的计算方式。若运送 aj 和运送 bk 是 v 的两个连续的运送,且运送 aj 先开始,约束(15)保证运送 bk 开始前 v 有足够的时间完成运送 aj 并行驶到 I/O 输送站 b 。约束(16)强制令 C_{\max} 等于 C_{aj} 的最大值。

(4) 运送起始位置约束

$$P_{aj}^s = P_a \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (17)$$

$$P_{aj}^c = \sum_{e=1}^N q_e w_{aje} \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (18)$$

$$C_{aj} - R_{aj} \geq (P_{aj}^c - P_{aj}^s)/V +$$

$$2u \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (19)$$

$$C_{aj} - R_{aj} \geq (P_{aj}^s - P_{aj}^c)/V +$$

$$2u \quad a = 1, \dots, m; j = 1, \dots, n_a \quad (20)$$

约束(17)强制令运送 aj 的开始位置为 I/O 输送机 a 的位置。约束(18)保证运送 aj 的完成位置为物料 J_{aj} 所分配的 I/O 站的位置。约束(19)和约束(20)保证在运送 aj 开始后 v 有足够的时间完成该运送。

假设运送 aj 是分界点左侧区域中的任意运送,运送 bk 是分界点右侧的任意运送,则运送 aj 始终分配给 $v1$,运送 bk 始终分配给 $v2$ 。下面对分区约

束和分界点碰撞避免约束进行分析,以下所有约束的成立条件为 $a, b=1, \dots, m; j=1, \dots, n_a; k=1, \dots, n_b; a \neq b$, 或 $\forall a=b$ 时, $j \neq k$ 。

(5) 分区约束

$$P_{aj}^s \leq P_{bk}^s + M(2 - z_{aj1} - z_{bk2}) \quad (21)$$

$$P_{aj}^c \leq P_{bk}^s + M(2 - z_{aj1} - z_{bk2}) \quad (22)$$

$$P_{aj}^s \leq P_{bk}^c + M(2 - z_{aj1} - z_{bk2}) \quad (23)$$

$$P_{aj}^c \leq P_{bk}^c + M(2 - z_{aj1} - z_{bk2}) \quad (24)$$

约束(21)至(24)确保 v_1 的所有运送的开始位置和完成位置都不大于 v_2 的任意运送的开始位置和完成位置。此时, v_1 和 v_2 的所有运送路线不会产生交叉。

(6) 区域分界点碰撞避免约束

$$R_{bk} \geq R_{aj} + u + (P_{aj}^s - P_{bk}^s + \theta)/V + (z_{aj1} + z_{bk2} + y_{aj,bk}^s - 3)M \quad (25)$$

$$R_{bk} - C_{aj} \geq (P_{aj}^c - P_{bk}^s + \theta)/V + (z_{aj1} + z_{bk2} + y_{aj,bk}^c - 3)M \quad (26)$$

$$C_{bk} \geq R_{aj} + 2u + (P_{aj}^s - P_{bk}^c + \theta)/V + (z_{aj1} + z_{bk2} + y_{aj,bk}^c - 3)M \quad (27)$$

$$C_{bk} \geq C_{aj} + u + (P_{aj}^c - P_{bk}^c + \theta)/V + (z_{aj1} + z_{bk2} + y_{aj,bk}^c - 3)M \quad (28)$$

结合约束(21), 约束(25)保证当运送 bk 的开始位置与运送 aj 的开始位置相同且 aj 先于 bk 开始的情况下, 运送 aj 开始后有足够的装载物料并行驶过 θ 距离后, v_2 才能到达该位置。类似地, 结合约束(22), 约束(26)确保当运送 bk 的开始位置与运送 aj 的完成位置相同且 aj 完成后 bk 才开始时, v_1 和 v_2 不会在该位置发生碰撞。结合约束(23), 可知约束(27)保证当运送 bk 的完成位置与运送 aj 的开始位置相同且运送 aj 开始后 bk 才完成时, v_1 和 v_2 不会在该位置发生碰撞。结合约束(24)可知, 约束(28)确保当运送 bk 的完成位置与 aj 的完成位置相同且 aj 完成后 bk 才完成时, v_1 能够及时避免与 v_2 在该位置碰撞。

2.3 目标函数

2-RGV 调度问题的目标为最小化物料的总出库时间, 目标函数表示为:

$$\text{Min } C_{\max}。$$

3 算法设计

虽然遗传算法全局搜索能力很强,但在局部搜索方面表现较弱,因此本文将遗传算法与局部改进算法相结合,提出一个混合遗传算法(HGA)求解 2-

RGV 分区调度问题。由于要为每件物料分配 RGV 和 I/O 站,因此,本文提出一种基于构建可行序列的染色体编码方式,以使染色体中的每个基因都包含运送编码、该运送所分配的 RGV 即 I/O 站的信息。下面给出 HGA 的详细过程。

3.1 染色体编码

基于构建可行序列的染色体编码方式的实施步骤如下。

步骤 1 根据物料的存储货格号对所有运送进行自然数编码。任意的 b 货架($b > 1$) k 货格里的物料对应运送的编码为 $\sum_{a=1}^{b-1} n_a + k$ 。染色体长度为出库物料总数。

步骤 2 RGV 的分配原则是平衡两辆 RGV 的运送量及减少两辆 RGV 的碰撞。由于对存取系统进行分区即是对所有货架和 I/O 站进行分区,因此本文对 m 排货架分区并进行车辆分配。记临界货架为 F ,则货架 1 至货架 F 作为区域 1,货架 $F+1$ 至货架 m 记为区域 2。首先将 F 的初始值设为 $\lfloor m/2 \rfloor$ 。比较区域 1 内的运送总量与区域 2 内的运送总量,若不相等,则调整 F 的位置。调整 F 的目的是减少两辆 RGV 运送量的差值, F 调整的停止准则为:差值为零,或差值不能再继续减小。当确定好临界货架后,根据每个基因中的运送编码解码出对应物料所在的货架编号,比较该货架编号与临界货架编号的大小以确定其所分配的 RGV。

步骤 3 由于 RGV 行驶时间占总出库时间很大比例,因此,本文采用贪婪策略对 I/O 站进行分配,即任意货架 a 上的物料总是被搬运至距离货架 a 最近的 I/O 站。

至此,一条代表运送序列、RGV 分配和 I/O 站分配的染色体构建完成。

3.2 初始种群的生成

为了保持种群的多样性,本文使用随机生成的方式构建染色体中代表运送序列的部分,代表 RGV 分配与 I/O 站分配的部分则按照上述完整染色体的构建方法构建。

由于问题的目标是最小化 C_{\max} ,因此本文的适应值函数的计算方式如下:计算每一代中每个个体对应的 C_{\max} ,找出其中最大的 C_{\max} 并记为 MC ,则个体的适应值函数可表示为 $Fitness = MC - C_{\max}$ 。

3.3 遗传算子的设计

(1) 选择操作。采用锦标赛选择策略,随机选择

种群规模的 $1/10$ 个染色体,从中选取适应值最大的个体作为一个父代染色体。依此方法再选出一个父代染色体,若两个染色体相同,再继续选取,直至所选的染色体与第一个选出的父代染色体不同为止。

(2)交叉操作。依照一定的交叉概率采用两点交叉策略进行交叉。随机生成两个基因位,并交换两个染色体对应的基因段,调整两条染色体的其余基因段中发生重复的基因。此时产生一对新染色体。

(3)变异操作。本文同时采用了反转变异和两点交换变异。当随机产生的数小于一个概率 δ 时,采用两点交换变异;当随机产生的数大于 δ 小于变异概率时,采用反转变异。反转变异是将随机产生的两个基因位之间的基因段顺序反转,两点交换变异是交换随机产生的两个基因位上的基因。

3.4 局部改进算法

局部改进算法是基于迭代邻域搜索的过程,即子代的每一个个体都经过一定次数的邻域搜索,并被其中最好的邻域染色体所替换。邻域的构造方法为随机交换染色体的两个基因位。

遗传算法的进化代数数和适应值的稳定度作为停止准则。当进化到指定代数或进化连续一定代数中,最优适应值没有改进时,遗传算法终止。

算法的实施步骤。

步骤1 初始化。设定遗传算法的参数。

步骤2 创建初始种群。

步骤3 计算种群中个体的适应值,并使用精英保留策略保留指定数量的染色体到下一代中。

步骤4 进行选择、交叉和变异操作,生成两个子代染色体。重复步骤4直至子代染色体数量达到种群数量。

步骤5 对子代种群中的每个个体应用局部改进算法,并记子代种群为当前种群。重复步骤3~步骤5,直至满足遗传算法的停止准则。

4 算例分析

4.1 参数设置

为了检验本文提出的 MILP 模型和 HGA 的有效性,本节给出了 12 组不同规模的算例进行测试。所有算例所使用的仓库布局的参数均来自某救灾物资 AS/RS。由于本文假设堆垛机拣取相邻货格中物料所需时间的间隔恒定,即,堆垛机拣取任意货格 k 里物料 J_{ak} 的时间为固定时间间隔 σ 的 $k-1$ 倍数加上 r_{a1} , $r_{ak} = \sigma(k-1) + r_{a1}$ 。12 组算例中, V 、 u 、 r_{a1} 和 σ 的取值都在实际应用取值范围内随机生成,详见表1。表2给出了算例规模的参数,其中前5个算例只使用了1-4号 I/O 输送机和1-2号 I/O 站。为了检测算法和模型的适用性,本文对每组算例中每排货架上待出库物料数做了不同设置。表3给出了12组算例的遗传算法参数设置。所有算例分别由 CPLEX12.5 和混合遗传算法分别进行求解。所有计算均使用 C++ 语言编程实现,在 3.10 GHz、4GB RAM 的 PC 上进行。

表1 AS/RS 系统参数设置

参数	取值
P_a (m)	0,3,6,9,12,15,18,21
q_e (m)	3,6,15,18
V (m/s)	$\sim U[1,3]$
u (s)	$\sim U[6,10]$
r_{a1} (s)	$\sim U[15,22]$
σ	$\sim U[1.5,3]$

表2 算例规模的设置

算例	m	N	n_a	NUM
小规模算例	S1	4	2	1,1,1,3
	S2	4	2	2,1,1,3
	S3	4	2	2,2,2,2
	S4	4	2	3,3,3,3
中等规模算例	M1	4	2	4,4,4,4
	M2	8	4	3,2,3,4,3,4,5,4
	M3	8	4	4,4,3,6,7,3,5,8
	M4	8	4	10,8,10,5,10,3,5,9
大规模算例	L1	8	4	12,8,7,4,11,5,9,6
	L2	8	4	9,9,9,9,9,9,9,9
	L3	8	4	10,10,10,10,10,10,10,10
	L4	8	4	12,12,12,12,12,12,12,12

表 3 遗传算法参数设置

参数	取值
种群规模	10, 10, 20, 30, 30, 50, 60, 120, 120, 120, 150, 160
遗传算法迭代次数	10, 10, 10, 20, 20, 30, 40, 50, 60, 60, 70, 70
无改进的代数	3, 3, 3, 5, 5, 10, 10, 15, 20, 20, 20, 20
交叉概率	0.7
变异概率	0.1
δ	0.05
精英保留数量	2, 2, 2, 3, 3, 4, 4, 6, 8, 10, 10, 15, 16
局部改进次数	2, 2, 2, 3, 3, 5, 6, 10, 10, 10, 10, 10

4.2 结果分析

对每组算例进行 10 次随机试验,记录 HGA 得

出的最好解、解的平均值及平均计算时间。同时,使用 CPLEX12.5 求解每组算例对应的 MILP 模型以得到每组算例的最优解。然而,由于问题的性质及规模的原因,只有小规模算例和中等规模中的算例 M1 能在 24h 内求出最优解。对于不能在 24h 内求出最优解的算例,依据问题规模设定一个计算时间,保存 CPLEX 在该时间内的解为 CPLEX 的最好解。表 4 给出了算例的计算结果,其中 HGA 在每组算例解的平均值 C_{HGA} 与 CPLEX 求解结果 C_{CPLEX} 的偏差的计算方式为 $gap = \frac{(C_{HGA} - C_{CPLEX})}{C_{CPLEX}}$ 。

从表 4 可以看出,CPLEX 能够精确求解规模较小的 5 组算例,对于这些算例,遗传算法可以求出最

表 4 计算结果对比

算例	MILP		HGA			gap(%)
	最优/最好解(s)	时间(s)	最好解(s)	平均值(s)	平均时间(s)	
S1	74.5	2.24	76.7	76.7	0.28	2.95
S2	83.5	2.46	83.5	83.5	0.29	0
S3	79.5	3.15	79.5	79.5	0.69	0
S4	115.5	13.39	115.5	115.95	3.04	0.39
M1	187.75	842.23	187.75	188.58	4.81	0.44
M2	360.5	*	348.5	350.3	26.97	-2.83
M3	551	*	507	509.3	72.23	-7.57
M4	1002.5	**	745	751.15	398.94	-25.07
L1	720.3	**	557	563.8	503.38	-21.73
L2	734.8	**	706.7	711.97	955.77	-3.11
L3	1013.9	**	768.3	777.21	839.83	-23.35
L4	1156.7	**	1035	1047.2	1269.15	-9.47

优解,或得出与最优解偏差较小的解。可以认为对于较小规模算例,HGA 能够得到问题的最优解或质量较好的近似最优解。对于规模较大的算例,CPLEX 只能在一定的时间内求出原问题的一个可行解,而 HGA 求得的解普遍比 CPLEX 的求解质量好。通过对比 CPLEX 求解 MILP 模型的时间和 HGA 求解算例所用的平均时间,可知遗传算法的平均求解时间较 CPLEX 的求解时间短,尤其在中等规模和大规模算例上,HGA 具有较强的优势。

为了说明本文的求解结果的确是无缝碰撞的调度,本文以算例 M2 为例,给出算例 M2 的 HGA 最好解,并给出其对应的时间-运送图,详见表 5 和图 3。图中横轴表示物料出库所用时间,纵轴表示 I/O 输送机和 I/O 站的坐标,虚线表示 RGV 的空运送或等待,实斜线表示 RGV 运送,位于某个坐标上的横实线表示 RGV 装载或卸载的动作。

综上所述,本文提出的 2-RGV 分区调度问题的混合整数线性规划模型能够有效避免 RGV 碰撞,

表 5 算例 M2 的 HGA 求解结果

出库顺序	$J_{11}, J_{71}, J_{82}, J_{12}, J_{13}, J_{64}, J_{22}, J_{84}, J_{21}, J_{83}, J_{32}, J_{72}, J_{43}, J_{33}, J_{63}, J_{74}, J_{42}, J_{73}, J_{31}, J_{41}, J_{61}, J_{62}, J_{44}, J_{81}, J_{52}, J_{75}, J_{51}, J_{53}$
所属车辆	1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1
对应 I/O 站	1, 4, 4, 1, 1, 3, 1, 4, 1, 4, 2, 4, 2, 2, 3, 4, 2, 4, 2, 2, 3, 3, 2, 4, 3, 4, 3, 3
运送开始时间(s)	17, 17, 38.5, 40, 63, 63, 84.5, 86, 104.5, 109, 126, 130.5, 147.5, 169, 169, 190.5, 190.5, 211.5, 212, 233.5, 233.5, 253.5, 256.5, 276.5, 281, 298, 304, 327

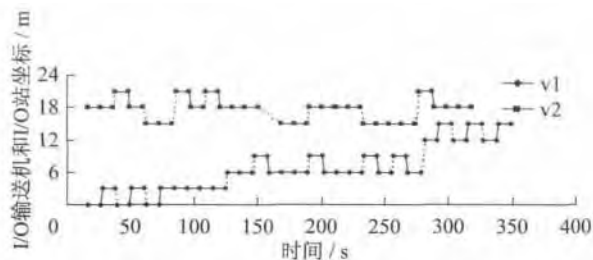


图3 算例 M2 的 HGA 求解结果的时间-运送图

所提出的混合遗传算法能够在较短时间有效地求解 2-RGV 分区问题。

5 结论

本文以某救灾物资立体仓库为背景,研究了出库过程中的 2-RGV 调度问题。通过将存取系统沿 RGV 轨道上某点划分为两个区域,并将两个区域内的运送分别分配两个 RGV 的方法减少两辆 RGV 之间的冲突。本文首先建立了分区域情况下的冲突避免的 2-RGV 调度问题的混合整数线性规划模型。其次,根据分区思想,采用了包含运送顺序、RGV 分配和 I/O 站分配信息的染色体编码方式,并提出一个混合遗传算法求解该问题。算例试验的结果表明本文提出的混合整数线性规划模型和遗传算法能够求解该问题。此外,虽然本文只研究了救灾物资立体仓库的出库过程中的 2-RGV 调度问题,但由于本文中的堆垛机速度、RGV 速度等参数都是根据实际取值范围随机抽取,并且所有算例中货架上待出库物料的分布有均匀和不均匀的情况,因此,本文提出的混合整数线性规划模型和求解方法对其他自动化立体仓库也具有借鉴意义。

参考文献:

- [1] 赵峰. 往复式穿梭车在 AS/RS 系统中的调度分析[J]. 起重运输机械, 2012(8): 58-61.
- [2] Lee S G, De Souza R, Ong E K. Simulation modeling 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.
- [3] Lee J, Maneesavet R. Dispatching rail-guided vehicles and scheduling jobs in a flexible manufacturing system[J]. International journal of production research, 1999, 37(1): 111-123.
- [4] Dotoli M, Fanti M P. A colored 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.
- [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. Deadlock detection and avoidance strategies for automated storage and retrieval systems[J]. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2007, 37(4): 541-552.
- [7] Liu Y, Li S, Li J, et al. Operation Policy Research of Double Rail-Guided Vehicle Based on Simulation[C]// E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference on IEEE, 2010: 1-4.
- [8] 吴长庆, 罗键, 陈火国等. 基于 Petri 网的 RGV 系统中环路死锁研究[J]. 计算机科学, 2009, 36(4): 250-253.
- [9] 杨少华, 张家毅, 赵立. 基于排队论的环轨多车数量与能力分析[J]. 制造业自动化, 2011, 33(16): 102-104.
- [10] 吴焱明, 刘永强, 张栋, 等. 基于遗传算法的 RGV 动态调度研究[J]. 起重运输机械, 2012(6): 20-23.
- [11] Lei L, Wang T J. The minimum common-cycle algorithm for cyclic scheduling of two material handling hoists with time window constraints[J]. Management Science, 1991, 37(12): 1629-1639.
- [12] Zhou Z, Li L. A solution for cyclic scheduling of multi-hoists without overlapping[J]. Annals of Operations Research, 2009, 168(1): 5-21.
- [13] 周支立, 李怀祖. 无重叠区的双抓钩周期性排序问题的求解[J]. 运筹与管理, 2006, 15(2): 1-7.



论文写作，论文降重，
论文格式排版，论文发表，
专业硕博团队，十年论文服务经验



SCI期刊发表，论文润色，
英文翻译，提供全流程发表支持
全程美籍资深编辑顾问贴心服务

免费论文查重：<http://free.paperyy.com>

3亿免费文献下载：<http://www.ixueshu.com>

超值论文自动降重：http://www.paperyy.com/reduce_repetition

PPT免费模版下载：<http://ppt.ixueshu.com>
