

基于磁悬浮列车的运行图编制算法

张其亮^{1,2}, 陈永生¹

(1. 同济大学 计算机科学与技术系, 上海 200331; 2. 江苏科技大学张家港校区 信息学院, 江苏 张家港 215600)

(ZQL6661979@163.com)

摘要:分析了轮轨列车与磁悬浮列车运行控制系统的差异,在参考轮轨列车运行图编制系统的基础上,提出了基于磁悬浮列车的运行图编制约束模型和编制算法。通过构建列车运行“位置-时间”矩阵检测列车运行冲突,并利用迭代修复算法消除冲突,最终得到可行的列车运行图。算例表明,该算法能有效地解决磁悬浮列车运行图编制方案,算法具有很好的有效性和实用性。

关键词:磁悬浮列车;列车运行图;分区控制;列车运行约束

中图分类号: U292.41; TP301.6 **文献标志码:** A

Working diagram algorithm of maglev train

ZHANG Qi-liang^{1,2}, CHEN Yong-sheng¹

(1. Department of Computer Science and Technology, Tongji University, Shanghai 200331, China;

2. School of Information, Zhangjiagang Campus of Jiangsu University of Science and Technology, Zhangjiagang Jiangsu 215600, China)

Abstract: The paper analyzed the difference of operation control system between railway train and maglev train. Based on the construction of the train working diagram of railways, the restriction model and algorithm of maglev train working diagram were put forward. The paper used the "station-time" matrix to check the train operation conflict and applied the iterative repair method to remove the conflicts; at last, the system got the feasible train working diagram. The computational example indicates that the proposed algorithm is able to work out the working diagram schemes of maglev effectively, and it has higher effectiveness and practicability.

Key words: maglev train; train working diagram; section control; train operation restriction

0 引言

列车运行图是行车工作的基础,运行图的优劣直接影响到运输的效率和安全性。目前的列车运行图大都是利用计算机自动编制。编图人员根据列车运行控制的特点,建立列车运行的数学模型并设计算法,借助于计算机对数学模型进行求解,最终得到可行的列车运行图。

国内外对计算机编制列车运行图的研究已经取得了一定的成果,建立了较为丰富和完善的数学理论和计算方法^[1-4],但是这些研究主要针对铁路、地铁等轮轨列车,对基于磁悬浮列车的计算机运行图编制研究还很少。磁悬浮列车的运行控制结构与传统轮轨列车存在一定的差异,不能直接把轮轨列车运行图的生成方法应用到磁悬浮列车,因此,借助轮轨列车运行图编制方法,进一步研究磁悬浮列车运行图的计算机编制具有重要的理论意义和实用价值。

本文根据磁悬浮列车运行控制的特点,建立了列车运行图的数学模型。提出首先建立松弛列车运行图^[5],通过构建“位置-时间”矩阵检查运行图中的冲突节点,利用迭代修复算法修复冲突的方法,最终生成了可行的列车运行图。

1 磁悬浮列车运行控制系统特点

磁悬浮列车的运行控制与传统轮轨列车的运行控制不同,它有特殊性,主要表现如下。

1) 磁悬浮列车采用分区控制。分区运行控制系统是高

速磁悬浮列车系统的核心中枢,完成列车驾驶控制、列车控制与防护、列车位置监控、进路及道岔控制与防护等功能^[6]。在分区的划分上磁悬浮列车与传统轮轨列车不同,按照运行需要将磁悬浮列车线路划分为若干分区,分区可以以车站为边界,也可以是线路上任一位置,且上行线和下行线的分区分别进行设置,这样使得磁悬浮列车线路结构更为复杂。普通轮轨列车分区是以车站中心线为边界的。

2) 磁悬浮列车在分区运行时,除特殊情况外,不能在分区上任意停车,而必须停靠在区段起始处的停车点上^[7]。

3) 磁悬浮列车线路中的一个分区只允许有一辆列车运行。传统轮轨列车在运行时一个分区允许有多辆列车运行,只要满足一定的追踪时间间隔即可。

磁悬浮列车分区控制系统与传统轮轨列车控制系统的差异,使得在编制磁悬浮列车运行图时,不仅要能体现列车在车站的到、发情况,还要体现列车在分区运行的信息,处理更为复杂。

2 运行图属性的数学描述

本文假设磁悬浮线路为双线运行模式,上下行列车相对方向运行,互不干扰。磁悬浮列车线路主要由分区和车站构成,即: $L = (Sec, Sta)$, $Sec = \{R_0, R_2, \dots, R_{n-1}\}$, 表示线路共划分为 n 个分区, $Sta = \{S_0, S_2, \dots, S_{m-1}\}$, 表示线路中有 m 个车站。约定 $t(i)$ 表示列车在分区 i 的纯运行时间, $t(i, j)$ 表示列车从车站 i 到相邻分区 j 的运行时间, $t(j, i)'$ 表示列车从分区

收稿日期:2010-09-15;修回日期:2010-11-07。 基金项目:国家“十一五”科技支撑计划项目(115-05-YK-038)。

作者简介:张其亮(1979-),男,山东潍坊人,讲师,博士研究生,主要研究方向:城市轨道交通、智能算法; 陈永生(1966-),男,江苏镇江人,研究员,博士生导师,主要研究方向:分布式系统、城市轨道交通。

j 到相邻车站 i 的运行时间。 tqi, tsi 分别表示列车在车站 i 的起车附加时分和列车在车站 i 的停车附加时分。 $SS^i(i)$ 、 $ST^i(i)$ 分别为列车在车站 i 的出发和到达时间。 $STO(i)$ 为列车在车站 i 的停站时间。 Us, Ust 为列车运营开始时间和列车运营结束时间。 $r^k(t)$ 表示列车 k 的始发时刻,始发时刻应限制在 $[Us, Ust]$ 范围内。 $Q^k(i)$ 、 $Q^k(i)'$ 分别表示第 k 次列车进入分区 i 和离开分区 i 的时间。

3 磁悬浮列车运行图编制模型

3.1 列车始发布点

运行图编制时首先要确定列车在始发车站的始发时刻,始发时刻的确定,本文采用初始布点法。初始布点法的基本思想是:从列车运营时间开始以一定的时间间隔 P 进行均匀布点,直到列车的运营结束时间。考虑磁悬浮列车运行时一个分区只允许有一辆列车运行,因此初始布点时间间隔 T 的取值为:

$$P = \max_{i=0}^{n-1} \{t(i) + \tau_{i\text{运}}\} \quad (1)$$

其中 $\tau_{i\text{运}}$ 为列车到达和离开第 i 个分区的附加时间,包括前一列车释放进路、释放分区的时间与后一列车准备进路、安全制动的的时间等。

式(1)中列车始发间隔时间 P 取值为 n 个分区中,列车在分区的运行时间与列车在该分区运行时的附加操作时间之和的最大值。因此,当后一辆列车发车时前一列车已经离开该分区,保证列车在始发车时一个分区只有一辆列车运行。由式(1)可得列车在始发车站的始发时刻为:

$$IS = \{Us, Us + P, \dots, Us + P * ((Ust - Us)/P - 1)\} \quad (2)$$

3.2 列车在中间车站/分区到发时间推算

根据列车在线路中运行时已设定好的列车在各站的停站时间、列车在分区的运行时间、列车起车附加时间、列车停车附加时间以及列车始发时间,按照列车经由的分区顺序和车站在分区中的位置可以推算出列车经由各中间车站和分区的到达、发车时刻。

对任一车站 $S_i \in Sta$,车站 S_i 所在分区 R_j 。

1) 车站在分区边界。

假设车站 S_i 的车站中心线为分区 R_j 的边界,则列车在该车站的到达时间和出发时间的递推关系为:

$$ST^i(i) = r^d(t) + \sum_{l=0}^{j-2} (t(I) + \tau_{l\text{运}}) + \sum_{k=0}^{i-2} (tqk + tsk + STO(k)) \quad (3)$$

$$SS^i(i) = ST^i(i) + STO^i \quad (4)$$

若车站在分区边界,则根据列车的始发时刻 $r^d(t)$ 通过式(3)、(4)即可得到列车在中间车站的到、发时刻。

2) 车站在分区内部。

假设车站 S_i 在分区 R_j 的内部,则列车在该车站的到达时间和出发时间的递推关系为:

$$ST^i(i) = r^d(t) + \sum_{l=0}^{j-2} (t(I) + \tau_{l\text{运}}) + \sum_{k=0}^{i-2} (tqk + tsk + STO(k)) + t(j-1, i)' \quad (5)$$

$$SS^i(i) = ST^i(i) + STO^i \quad (6)$$

若车站在分区内部,则根据列车的始发时刻 $r^d(t)$ 通过式(5)、(6)即可得到列车在中间车站的到、发时刻。

对于列车进入分区和离开分区的时刻可以利用同样的方

法推算。在推算时首先要对线路中的分区和车站按照列车运行(上行或下行)顺序进行排列,得到线路拓扑图,然后才能按照分区、车站之间的位置进行时间的推算。

3.3 磁悬浮列车运行图约束模型

根据磁悬浮列车运行控制的特点和列车运行要求,列车运行过程中的约束条件由以下模型确定:

$$Us \leq r^k(t) \leq Ust \quad (7)$$

$$Q^{k+1}(i) - Q^k(i)' - \tau_{\text{连}} \geq 0 \quad (8)$$

$$\text{Count} - N_{\text{车底}} \geq 0 \quad (9)$$

$$R^i - t_{\text{折返}} \geq 0 \quad (10)$$

式(7)用于限制列车发车时间,式(8)用于限制同方向列车连发时间,式(9)限制使用的列车车底数,式(10)限制列车折返时间。同时,列车运行时的时间间隔约束还包括会车时间间隔($\tau_{\text{会}}$)、不同方向列车的不同时发到时间间隔($\tau_{\text{不发到}}$)、同方向列车的不同时发到时间间隔($\tau_{\text{发到}}$)、同方向列车不同时到发时间间隔($\tau_{\text{到发}}$)。

4 运行图编制算法

列车运行图的铺画总体可以分为三个步骤:得到松弛列车运行图,检测松弛列车运行图中的冲突,迭代修复松弛列车运行图直至得到可行的列车运行图。运行图生成步骤如图1所示。

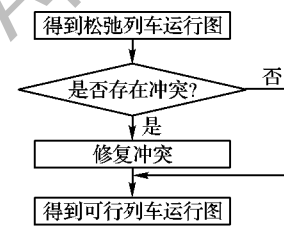


图1 运行图生成流程

松弛列车运行图是指初始列车运行图,通过始发时刻和列车在分区运行时间及车站停站时间即可得到,松弛列车运行图一般来说都会存在冲突^[5],必须能检测出冲突并能修复冲突。

4.1 运行图冲突检测方法

磁悬浮列车采用分区控制,其分区结构要比轮轨列车的分区结构复杂,不同的分区结构导致列车的运行约束不一致。在冲突检测时首先根据车站所在的分区结构判断列车在该站运行时应具有约束条件,然后根据列车的到、发时刻检验是否满足约束条件,不满足则表明存在冲突。除了在车站检测冲突外,还要在分区检测冲突,分区的冲突检测主要是检测是否满足式(8)存在的连发约束和一个分区仅允许一辆列车运行的约束。

为了便于查找松弛列车运行图中存在的冲突,本文将线路中所有的分区和车站按照其坐标由小到大的顺序进行排列,并利用矩阵结构将各次列车经过车站或分区时的时间信息表示出来,矩阵结构如图2所示。

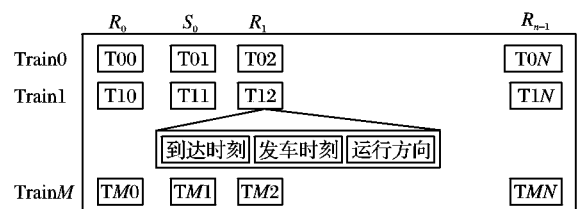


图2 列车“位置-时间”结构图

列车经过车站或分区时的信息用节点 T 描述,节点 T 是一个三元组,包含列车到达时刻、列车发车时刻和列车运行方向。如果列车不经过该分区或在车站不停车,则 T 中到、发时刻设置为 0。

通过建立的矩阵结构进行冲突检测的原理如下。

首先,按照列车运行顺序对矩阵纵向轴中每个时间节点与其前相邻时间节点进行比较,纵向轴代表所有列车在该分区或车站的到、发时刻。如果相邻两节点不满足约束条件则表明存在冲突,标记出冲突的节点。

然后,对矩阵横向轴中相邻两列车在同一分区的离开时间和到达时间进行比较,即:比较前一列车在分区的离开时刻与该列车在分区的到达时刻,比较的结构如图 3 所示,主要是对约束式(8)进行检查,不满足约束则标记节点。

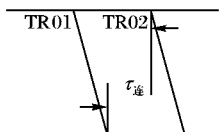


图3 连发间隔约束

按照上述原理,在矩阵结构中逐分区、逐车站进行比较,比较时先比较纵向轴中各时间节点,再横向比较相邻两列车在同一分区的到达和出发时刻,直到比较完所有分区和车站为止。通过上述方法可以找到运行图中存在冲突的所有节点。

4.2 迭代修复算法

运行冲突的解决方法主要是通过调整冲突节点中列车到达或发车时刻,使其满足时间间隔要求完成的。在解决冲突的过程中一个冲突的解决往往会导致一个或多个新的冲突的产生^[8]。为此,本文提出利用迭代修复算法经过多次迭代消除运行图中所有冲突。

修复冲突时按照列车经过分区、车站的顺序完成。按照图 2,首先解决分区 R_0 中的运行冲突,然后是 S_0, R_1, \dots, R_{n-1} 。随着冲突的解决,冲突逐步被移至后续列车和后续车站、分区,直至所有冲突全部消除。图 4 为列车在车站和分区的时间节点,空心节点为存在冲突的节点,实心节点为非冲突节点。迭代修复冲突算法基本思想如下。

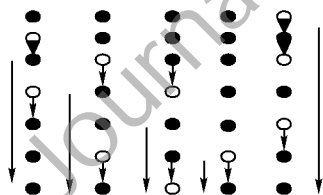


图4 列车时间节点图

按照图 4 所示的节点情况,首先从列车运行的第一列查找冲突节点,找到冲突节点后对节点中的到、发时刻进行调整,消除该节点与前相邻节点的冲突,消除该节点的冲突标志。然后判断该节点与后相邻节点是否存在冲突,如果存在冲突则标记后相邻节点为冲突节点,用同样的方法解决后续列车中的冲突节点。当第一次迭代完成后,进行下一次的迭代,同样的方法直至消除该分区或车站上的所有冲突节点,进入下一分区或车站用相同的方法进行处理。

迭代修复算法解决冲突的步骤如下。

- 1) 按照列车运行顺序,从第一个分区开始查找冲突节点。
- 2) 若当前分区或车站中存在冲突节点则转步骤 3), 否则

转步骤 4)。

3) 在当前分区或车站上找到冲突节点,调整冲突节点的到或发车时刻,使其满足时间间隔约束。消除该节点的冲突标志,判断该节点与后相邻节点是否存在冲突,是则标记后相邻节点为冲突节点,否则修复该分区或车站中的下一个冲突节点。第一次修复完成后,转步骤 5)。

4) 进入下一分区,检查是否存在冲突,是则转步骤 3), 否则判断该分区是否为最终分区,是则转步骤 6), 否则转步骤 4)。

5) 检查当前分区或车站上是否仍有冲突节点,若有则转步骤 3) 进行下一次迭代,否则转步骤 4)。

6) 输出调整后的可行列车运行图。

5 实验结果及分析

为检验算法的有效性,本文模拟了一条线路,线路中设置了 7 个分区,4 个车站。并在 VC++ 环境下进行了仿真,生成的部分运行图如图 5 所示。图中横坐标表示列车运行时间,纵坐标表示车站和分区的位置,由于磁悬浮列车以分区为单位进行控制,因此在运行图中将分区信息显示了出来。车站和分区的位置是根据车站和分区的坐标按照一定比例显示的。相邻两纵线之间的间隔表示 1 min,该运行图显示了 0 ~ 70 min 之间的列车运行信息。以运营开始时间为 0 时 0 分 0 秒,运行结束时间为 24 时 0 分 0 秒计算,在给分分区运行时间、停站时间和折返时间的前提下铺画列车运行图,经计算最少可用列车车底数为 4,列车发车间隔为 600 s 左右,生成运行图的时间为 0.031 s。经检验生成的列车运行图不存在冲突,能得到最大能力运行图,满足仿真要求,运行图生成算法有效可行。

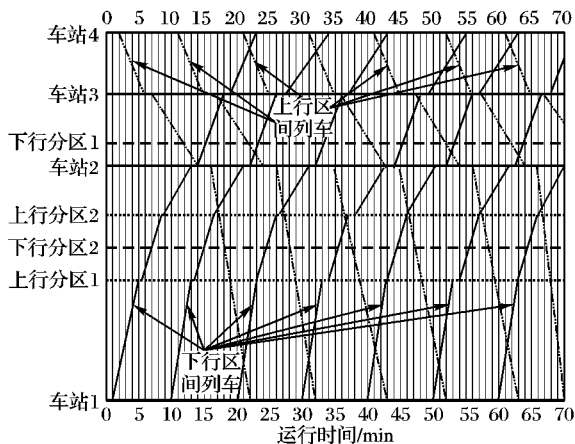


图5 列车运行图

6 结语

本文分析了磁悬浮列车运行控制系统的特点,并与普通轮轨列车的运行控制进行比较,得出了基于磁悬浮列车运行图约束模型,提出了一种基于矩阵结构进行冲突检测的方法,并利用迭代修复算法对冲突进行消除。通过实验仿真,证明了本文提出的冲突检测方法和迭代修复算法在查找运行图冲突和解决运行图冲突中具有良好的实用性。

参考文献:

- [1] 孙焰,李致中.单线区间列车最优运行次序的排序模型及解法[J].铁道学报,1993,15(1):62-71.

(下转第 825 页)

性。

本文采用的目标系统包含一个 ARM7TDMI 和一块现场可编程门阵列(Field Programmable Gate Array, FPGA),其中软件任务在 ARM7TDMI 上实现,硬件任务在 FPGA 上实现。同时,为简化起见,做如下合理的假设:

- 1) 任务在 CPU 和在 FPGA 上实现所需的运行时间和占用的硬件面积是静态的,可预先计算或者通过经验估算出来;
- 2) 不考虑同种实现中通信的开销;
- 3) 不考虑软/硬件任务的并行性。

3.1 求解质量对比

实验结果如表1所示,表中数据表示四种算法在不同节点数下找到最优解的个数,其中遗传算法和蚂蚁算法动态融合的算法(Dynamic Combination of Genetic Algorithm and Ant Algorithm, DCG3A)是文献[3]中采用的软/硬件划分算法。每组任务图具有30个不同的样本图形,将四种算法(包括TS、HNN、DCG3A、TS_HNN)均运行10次。对于固定属性的任务图,不同算法获得的最优解质量相差较小。因此,为了使比较更加公平,不直接比较解的质量,而是通过比较算法找到最优解的个数以判断其性能。

表1 四种算法在不同节点数下得到的最优解个数

算法	节点数							
	10	30	60	90	120	150	180	210
TS	299	290	280	277	260	236	172	170
HNN	297	286	270	253	230	204	170	160
DCG3A	299	296	285	263	265	234	216	201
TS_HNN	298	297	290	285	277	269	259	251

从表1中可看出,在节点数位10和30的两个实例中,四个算法的表现都较好,但是随着节点数增加,它们的表现差异较大。TS_HNN算法在实验中表现出比其他三种算法更加优秀的全局搜索能力,这应该归功于Hopfield神经网络快速求解以及TS算法较强的“爬山”能力,这两个特性可避免搜索过程陷入局部最优。

3.2 运行时间对比

与表1类似,对每组任务图将四种算法均执行10次,得到的平均运行时间如表2所示。

表2 四种算法在不同节点数下的运行时间 s

算法	节点数							
	10	30	60	90	120	150	180	210
TS	0.239	0.580	1.397	2.598	5.056	8.711	19.328	33.231
HNN	0.135	0.339	0.890	2.256	5.013	8.342	19.631	32.012
DCG3A	0.117	0.298	0.832	1.861	4.315	7.631	16.818	30.233
TS_HNN	0.134	0.335	0.900	1.761	4.221	6.532	12.710	20.125

从表2中可看出,当任务图节点数较少(10~60)时,TS_HNN算法略优于其他三种算法,但效果不很明显;而当节点数超过

60时,TS_HNN算法运行时间明显优于其他三种算法。同时,随着任务图规模的增大,TS_HNN算法性能改进的效果越显著,这是因为在TS_HNN算法中使用HNN算法能较快地获得局部最优解,从而使得TS算法搜索范围更加接近全局最优解,从而提高TS算法搜索速度。

4 结语

面向嵌入式软件功耗相关的软/硬件双路划分问题,本文提出一种基于离散Hopfield神经网络和禁忌搜索融合的求解算法,吸取了HNN算法与TS算法在寻优问题上各自的优势,克服了它们的不足。仿真实验证明,与同类算法相比,该算法在针对嵌入式软件功耗相关软/硬件划分问题的求解中具有较高的稳定性,且搜索速度也较快,随着问题规模的增大,性能的改进更加明显。下一步需要研究的主要问题是多个约束条件下(包括执行时间、面积、成本等),对软件功耗进行优化。

参考文献:

- [1] 罗刚,郭兵,沈艳,等. 源程序级和算法级嵌入式软件功耗特性的分析与优化方法研究[J]. 计算机学报, 2009, 32(9): 1869 - 1875.
- [2] MICHAEL R G, DAVID S J. Computers and intractability: A guide to the theory of NP-completeness [M]. San Francisco: W H Freeman & Co Ltd, 1979: 145.
- [3] 熊志辉,李恩昆,陈吉华. 遗传算法与蚂蚁算法动态融合的软硬件划分[J]. 软件学报, 2005, 16(4): 503 - 512.
- [4] DICK R P, JHA N K. MOGAC: A multiobjective genetic algorithm for the co-synthesis of hardware-software embedded systems [C]// IEEE/ACM International Conference on Computer-Aided Design. San Jose: IEEE Computer Society, 1997: 522 - 529.
- [5] JORG H. A low-power hardware/software partitioning approach for core-based embedded systems [C]// Proceedings of the 36th ACM/IEEE Conference on Design Automation. New Orleans, LA: IEEE, 1999: 122 - 127.
- [6] 马天义,刘宏伟,温东新,等. 面向多处理器 SoC 设计的低功耗软硬件划分[J]. 高技术通讯, 2007, 17(10): 991 - 996.
- [7] GUO BING, WANG DIANHUI, SHEN YAN, et al. Hardware-software partitioning of real-time operating systems using Hopfield neural networks [J]. Neurocomputing, 2006, 69(16/17/18): 2379 - 2384.
- [8] PEREIRA C, RAGHUNATHAN V, GUPTA S. A software architecture power aware real time operating systems, GECS Technical Report 02-02 [R]. Irvine: University of California, 2002.
- [9] 石敏. 基于遗传和禁忌搜索混合的 SoC 软硬件划分方法研究[D]. 哈尔滨: 哈尔滨理工大学, 2009.
- [10] HOPFIELD J J, TANK D W. Neural computation of decisions in optimization problems [J]. Biological Cybernetics, 1985, 52(3): 141 - 152.

(上接第814页)

- [2] GHOSEIRI K, SZIDAROVSKY F, ASGHARPOUR M J. A multi-objective train scheduling model and solution [J]. Transportation Research Part B, 2004, 38(10): 927 - 952.
- [3] 许红,马建军,龙建成,等. 客运专线列车运行图编制模型及计算方法研究[J]. 铁道学报, 2007, 29(2): 1 - 7.
- [4] 史峰,黎新华,秦进,等. 单线列车运行图铺划的时间循环迭代优化方法[J]. 铁道学报, 2005, 27(1): 15 - 21.
- [5] 周文梁,史峰,陈彦. 基于定序优化的客运专线列车运行图铺划

方法[J]. 铁道学报, 2010, 32(1): 1 - 7.

- [6] 徐洪泽,郑伟,刘湘黔,等. 高速磁浮列车分区运行控制系统的设计与实现[J]. 高技术通讯, 2007, 17(9): 958 - 961.
- [7] 张振江,刘云,冯玉琨. 磁悬浮列车追踪模型的研究与仿真[J]. 系统仿真学报, 2006, 18(4): 841 - 846.
- [8] 章优仕,金炜东. 基于离散事件拓扑图模型的单线列车运行调整迭代修复算法[J]. 系统仿真学报, 2009, 21(11): 7003 - 7007.