

# 多资源约束下的项目调度邻域搜索算法<sup>①</sup>

张 凯

(浙江工业大学 教育科学与技术学院, 杭州 310023)

**摘 要:** 为了解决项目调度过程中不同项目之间的资源冲突问题, 提出了一种基于拓扑排序的邻域搜索算法. 设计跳动和微动两个操作, 实现编码的更新, 保证解的多样性与收敛性. 用项目调度问题库(Project Scheduling Problem Library-PSPLIB)中不同规模的实例对算法进行检验, 结果表明, 这种邻域搜索算法能够找到较优的解. 用这种邻域搜索算法计算其它算法的案例, 结果也显示这种算法具有一定的优越性. 对这种邻域搜索算法的不同参数进行实验, 分析了参数设置对算法性能的影响.

**关键词:** 拓扑排序; 跳动; 微动; 邻域搜索

## Neighborhood Search Algorithm to Solve Resource Constrained Project Scheduling Problem

ZHANG Kai

(College of Educational Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract:** To solve the resource conflicts between different activities, a new neighborhood search algorithm based on topological sorting was proposed. Two operations namely jumping move and micro-move were designed to update code and get better results. Use cases with different scales from Project Scheduling Problem Library to test the performances of the algorithm. The results showed that the algorithm had searched excellent solutions. Use the algorithm proposed to solve the case in the literature, the results showed that the algorithm had its superiority. Test the proposed algorithm with different parameters and analyze the influences of parameter setting on algorithm performance.

**Key words:** topological sorting; jumping move; micro-move; neighborhood search algorithm

### 1 引言

资源受限项目调度问题(Resource-constrained Project Scheduling Problem, RCPS)是指在满足项目时序要求和多种资源的约束下, 确定所有任务的开始时间及结束时间, 使得整个项目的完工时间最短. 资源受限项目调度问题广泛存在于飞机制造、软件开发、工程建筑等领域. 合理地调节各项目活动的开始时间和资源能够有效地缩短项目完工时间, 降低成本. 在实际生产中具有非常重要的意义.

RCPS 问题属于 NP-hard 问题<sup>[1]</sup>, Erick<sup>[2]</sup>等人用分支定界法来求解资源限制下的项目调度问题; Patterson<sup>[3]</sup>等人用整数规划的方法求解 RCPS. 这两种方法都能够找到问题的最优解, 但也都存求解问题

规模小, 求解速度慢的缺点. 敬石开<sup>[4]</sup>等人提出了多项目总工期及各项目工期加权和最小的数学模型, 提出了基于贪婪策略的求解方法; 刘金定<sup>[5]</sup>等人提出了启发式串行算法求解 RCPS; 廖仁<sup>[6]</sup>等人提出了 4 条调度项目活动的规则. 以上三种方法属于启发式算法, 能够对特定的问题求得较好的解, 但是算法的通用性较差. Jedrzejowicz<sup>[7]</sup>等人提出了因子组解法求解 RCPS; 应瑛<sup>[8]</sup>等人提出了求解 RCPS 的混合遗传算法. 以上两种算法属于智能算法, 能够较快地找到问题的满意解, 但是遗传算子等操作设计过程繁杂.

本文提出了基于拓扑排序的一种邻域搜索方法, 操作比较简单, 解的搜索能力强, 能够很好地解决 RCPS.

<sup>①</sup> 基金项目: 2013 年浙江工业大学大学生创新创业训练计划项目(2013031)

收稿时间:2013-07-02;收到修改稿时间:2013-08-06

## 2 资源受限项目调度数学模型

一个项目包括  $n$  个活动, 其中活动 1 和活动  $n$  是两个虚拟的活动, 分别表示项目的开始和项目的结束.  $t_i$  表示活动  $i$  的开始时间,  $t_n$  表示整个项目的完工时间. 如果活动  $j$  和活动  $i$  之间有时序约束关系, 活动  $j$  必须在活动  $i$  完成之后开始, 那么活动  $i$  和活动  $j$  属于时序约束集合  $H$ .  $d_i$  表示活动  $i$  的持续时间. 资源种类为  $m$  种, 在每一时刻  $t$ , 需要处理的所有活动集合为  $A_t$ . 活动  $i$  需要资源  $k$  的量为  $r_{ik}$ . 资源  $k$  的总量为  $b_k$ , 每一时刻  $t$  资源的消耗量不应大于资源的总量. 项目调度的数学模型可以描述为:

$$\begin{aligned} & \min \quad t_n \\ & s.t. \quad \begin{cases} t_j - t_i \geq d_i, (i, j) \in H \\ \sum_{A_t} r_{ik} \leq b_k \\ t = 1, \dots, T, k = 1, \dots, m \end{cases} \end{aligned} \quad (1)$$

## 3 基于拓扑排序的邻域搜索算法

### 3.1 基于拓扑排序的编码

拓扑排序是图论中的一个概念, 是对有向无圈图的一种排序, 使得如果存在一条从顶点  $A$  到顶点  $B$  的路径, 那么在排序中  $B$  出现在  $A$  后面.

拓扑排序运用到项目调度中, 表示活动开始时间的先后顺序. 在项目调度中, 项目网络图表示了各个活动之间的前后关系. 图 1 是一组编码, 如果活动  $j$  必须在活动  $i$  完成后才能开始, 则编码中  $j$  必须出现在  $i$  之后. 可以从项目网络图得到项目各活动之间的邻接矩阵  $R$ , 根据邻接矩阵  $R$  得到项目各活动的拓扑排序.

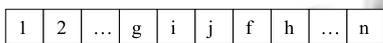


图 1 基于拓扑排序的编码

### 3.2 译码及目标函数

译码过程是根据项目各活动的拓扑排序  $TP$ , 所需的资源  $r$ , 时间  $d$ , 资源限制  $RT$  和邻接矩阵  $R$  求出项目各活动的开始和结束时间, 目标函数是最小化整个项目的完成时间. 译码步骤如下.

步骤 1: 对于拓扑排序中的第一项活动  $i$ , 在  $RT$  矩阵中找出资源  $k(k=1,2,\dots,m)$  所对应的行  $RT_k$ , 从  $RT_k$  找出活动  $i$  可以开始的时间  $t_{ik}$ , 活动  $i$  的开始时间  $t_i = \max(t_{ik})$ ,  $t_i + d_i$  作为活动  $i$  的结束时间, 更新  $RT$ .

步骤 2: 若所有活动的开始时间和结束时间都已计算出, 则结束, 项目完工时间为  $t_n$ . 否则, 进入步骤 3.

步骤 3: 对于下一项活动  $j$ , 从邻接矩阵  $R$  中找出必须领先于活动  $j$  的所有活动的完成时间, 从该时刻起往后找  $RT$  中满足活动  $j$  资源需求的时刻  $t_j$  作为活动  $j$  的开始时间,  $t_j + d_j$  作为活动  $j$  的完成时间.

步骤 4: 返回步骤 2.

### 3.3 终止条件

运用迭代总数  $N_d$  作为算法终止条件, 当算法迭代代数到达设定的  $N_d$  时, 程序停止.

### 3.4 种群选择

对于每次生成的新种群, 设计了精英策略来保护优良解. 设置精英集大小为  $N_j$ . 保留上一代中最好的  $N_j$  个个体进入下一代, 对于剩下的个体, 运用轮盘赌法选择.

### 3.5 跳动操作

跳动是指拓扑排序中的某些活动的位置移动较长距离, 从而导致拓扑排序结构变化, 达到更新编码的目的. 图 2 给出了一个编码的跳动操作.

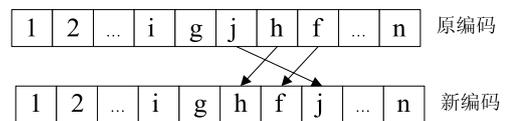


图 2 跳动操作

跳动过程中需要检验活动  $i$  能否从位置 1 跳到位置 2, 需要用可达矩阵  $D$  来判断.  $R$  为邻接矩阵,  $I$  是单位阵,  $D = (R+I)^N$ ,  $N$  的约束条件是  $(R+I)^N = (R+I)^{N+1}$ .

跳动操作的步骤如下.

步骤 1: 跳动操作的位置个数为  $N_{iao}$ , 随机产生  $N_{iao}$  个不重复的  $1-n$  的随机数, 作为可能进行跳动操作的活动的位置.

步骤 2: 对于一个可能跳动位置  $w_1$  对应的活动  $i$ . 如果该  $i=n$ , 则不进行操作; 否则, 从可达矩阵  $D$  中找出所有开始时间在活动  $i$  完成之后的活动集合.

步骤 3: 如果该活动  $i=n$ , 则不进行操作; 否则, 从可达矩阵  $D$  中找出所有开始时间在活动  $i$  完成之后的活动集合  $K_i$ .

步骤 4: 从  $w_{1+1}$  开始, 找出集合  $K_i$  中最先出现元素的位置  $w_2$ , 如果  $w_2 = w_1 + 1$ , 不进行操作; 否则, 把产品  $i$  从位置  $w_1$  移动到位置  $w_2 - 1$ . 位置  $w_1 + 1$  到位置  $w_2 - 2$

所对应的所有活动在编码中各向左移动 1 个单位。

步骤 5: 如果第  $N_{tiao}$  个位置已进行跳动操作, 则停止, 否则, 回到步骤 2。

### 3.6 微动操作

微动是指某些活动的位置与相邻的活动的位置进行互换, 从而达到更新编码的目的。图 3 中给出了一个点的微动操作。

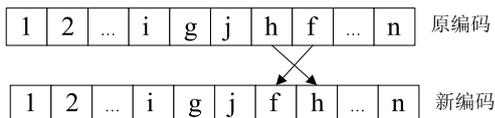


图 3 微动操作

微动操作的步骤如下。

步骤 1: 设定微动操作的位置个数  $N_{wei}$ , 随机产生  $N_{wei}$  个不重复的 1- $n$  的随机数, 作为可能进行跳动操作的活动位置。

步骤 2: 对于一个可能微动的位置  $w_1$  对应的活动  $i$ ; 如果位置  $w_1=1$ , 则不进行操作, 否则, 从可达矩阵  $D$  找出所有必须在产品  $i$  开始前结束的活动集  $K_2$ 。

步骤 3: 位置  $w_2 = w_1 - 1$ , 如果位置  $w_2$  上的活动  $j$  属于  $K_2$ , 则不进行微动操作, 否则, 活动  $i$  和活动  $j$  的位置进行互换。

步骤 4: 如果第  $N_{wei}$  个位置已经行微动操作, 则结束; 否则, 回到步骤 2。

### 3.7 基于拓扑排序的邻域搜索算法流程

图 4 给出了整个基于拓扑排序的项目调度邻域搜索算法的流程。

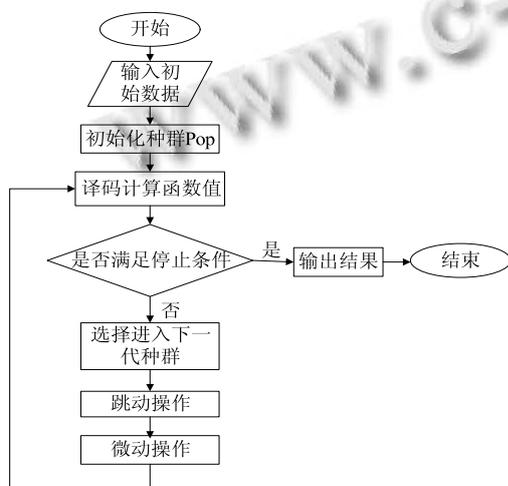


图 4 邻域搜索算法流程图

## 4 算例验证

项目调度问题库 (Project Scheduling Problem Library-PSPLIB) 是一个专门用来验证项目调度算法性能的标准实例库, 它的实例由标准项目产生软件 ProGen 产生。它是一个开元的实验库, 随着研究的深入, 实例库的实例的解也在不断地更新。

本文将采用该标准实例库中活动规模大小为 30, 60, 90 项活动的实例各 3 例进行算法性能的验证。不失一般性地在 J30 的实例中选择了实例 J301\_1, J301\_2, J301\_3, 在 J60 的实例中选择 J601\_1, J601\_2, J601\_3, 在 J90 的实例中选择了 J901\_1, J901\_2, J901\_4 作为实验对象。所有实验的迭代代数都设置为 50, 精英集大小都为  $N_j=3$ , 种群大小  $popsiz$  设为 8。在 J30, J60, J90 规模的实验中, 跳动位置数分别  $N_{tiao}=2, 5, 10$ , 微动位置数分别设置为  $N_{wei}=2, 5, 10$ 。实验的种群大小  $popsiz=8$ 。参数的选择将在第 5 部分具体说明。图 5, 图 6, 图 7 分别给出了 J30, J60 和 J90 规模下, 实例的算法迭代图。红色的虚线给出了 PSPLIB 中最新的对应问题的解。从图 5 中可以看出规模为 32 个活动(包括首末两个虚拟活动)的 3 个实例都找到了实例库中当前的最好解。图 6 中显示 62 个活动的资源约束下项目调度问题的 3 个实例也找到了实例库中当前的最好解。J30 和 J60 问题解的收敛速度很快, 在迭代到 35 代左右时收敛到了最好解。图 7 中显示了 J90 下的 3 个实例求解的迭代图, 由于 J901\_2 和 J901\_3 结构相似, 用 J901\_4 替换 J901\_3 进行实验。J901\_2 和 J901\_4 都找到了最好解。对于 J901\_1, 本算法找出的解为 76, 离 PSPLIB 的最好解 73 相差也较近。表 1 给出了 J901\_4 的项目调度结果。

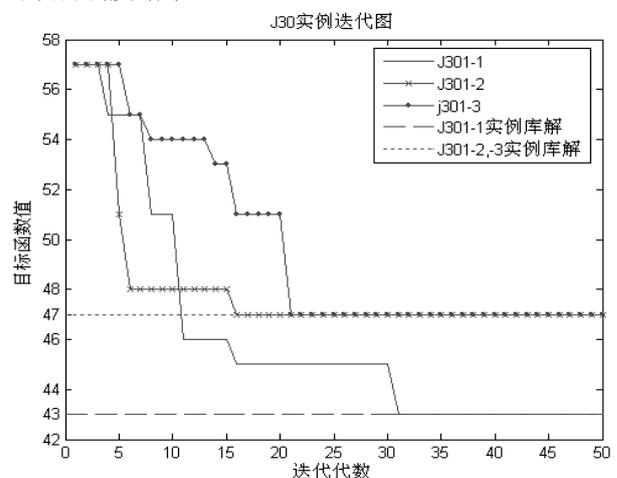


图 5 J30 实例迭代图

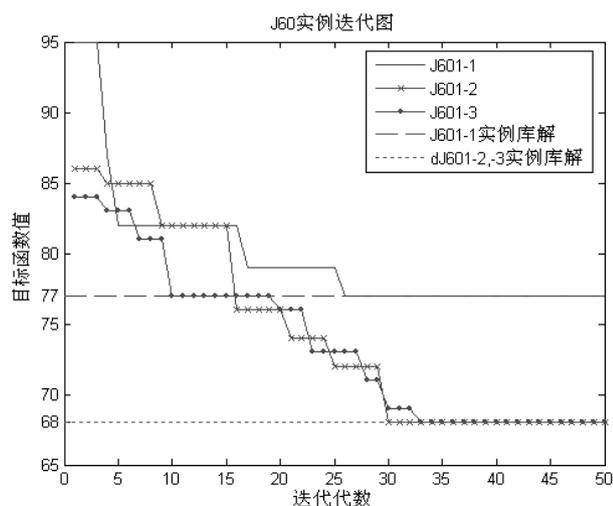


图 6 J60 实例迭代图

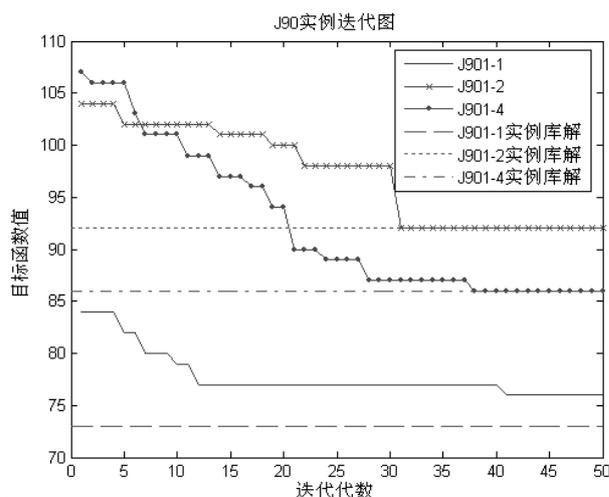


图 7 J90 实例迭代图

表 1 J901\_4 实例项目调度结果

No	St	Ft	No.	St	Ft	No.	St	Ft	No	St	Ft	No.	St	Ft	No	St	Ft
1	0	0	17	18	20	33	32	33	49	33	40	65	53	63	81	57	64
2	13	20	18	21	30	34	38	41	50	42	43	66	36	42	82	67	76
3	0	5	19	13	18	35	40	50	51	36	40	67	55	61	83	64	73
4	0	4	20	16	21	36	36	38	52	42	45	68	59	63	84	47	53
5	10	11	21	29	32	37	25	28	53	15	16	69	45	49	85	56	58
6	23	31	22	14	16	38	50	53	54	11	12	70	44	49	86	70	74
7	5	13	23	30	33	39	16	21	55	40	45	71	36	43	87	71	79
8	4	13	24	23	29	40	28	36	56	45	52	72	50	56	88	74	82
9	11	15	25	21	25	41	42	48	57	28	33	73	43	47	89	76	83
10	13	21	26	32	36	42	53	57	58	48	50	74	28	35	90	73	81
11	4	14	27	29	36	43	22	28	59	42	45	75	67	70	91	82	86
12	11	12	28	45	55	44	35	43	60	32	39	76	49	58	92	86	86
13	21	29	29	20	24	45	64	66	61	48	53	77	41	44			
14	21	22	30	16	23	46	49	59	62	33	38	78	58	62			
15	5	10	31	33	42	47	21	29	63	61	71	79	42	48			
16	13	18	32	55	64	48	14	17	64	58	67	80	50	57			

由 9 个算例可以看出本文提出的邻域搜索算法具有可行性, 以及具有较强的搜索优秀解的能力。

### 5 算法比较

为了检验本文提出的基于拓扑排序的邻域搜索算法的性能. 用本文提出的算法计算其它两种算法的案例. 两种算法分别是, 关键链识别算法, 基于贪婪策略的算法. 对于关键链识别算法的计算用到的实例是 PSPLIB 中的 J301\_1<sup>[9]</sup>. 对于基于贪婪策略的算法的实例由文献 4 给出.

图 5 中的第一条标示的实线给出了运用本文提出

的算法求解 J301\_1 问题的求解的迭代过程. 由图 5 看到本文提出的邻域搜索算法求得的最短项目工期为 43, 但是基于关键链识别算法求解的结果为 49<sup>[9]</sup>, 本算法在这个实例中求解结果要比基于共价键识别算法优.

对于文献 4 中的实例, 表 2 给出了文献中的方法以及本文提出的算法的计算结果.

表中黑体表示本文算法优于其它算法的数据. 可以看出本文提出的算法在项目总工期以及加权和上均优于文献中的算法.

从以上算法的比较中可以看出本文提出的基于拓扑排序的邻域搜索算法具有一定的优越性.

表 2 各种算法对文献[9]中的实例计算结果

算法	项目 1 时间	项目 2 时间	项目 3 时间	项目 4 时间	项目 5 时间	总工期	加权和
贪心策略算法 <sup>[9]</sup>	35	21	27	30	17	35	32.29
关键链识别 <sup>[10]</sup>	30	21	34	28	17	34	31.14
遗传算法 <sup>[11]</sup>	31	22	33	26	18	34	31.30
LSSA <sup>[12]</sup>	32	31	35	29	20	37	34.48
本文算法	<b>24</b>	22	31	30	21	<b>31</b>	<b>28.54</b>

## 6 参数分析

除了编码方式,解更新策略的设计之外,种群规模  $popsiz$ , 迭代数  $N_d$ , 跳动数  $N_{iao}$  和微动数  $N_{wei}$  等参数的选择是影响算法求解的另一些关键因素. 因此,以文献 4 的实例作为实验对象. 在总个体数( $popsiz \times N_d$ )相同的情况下,  $Popsiz$  分别设置为 5, 8, 10,  $N_d$  分别为 8, 5, 4, 跳动数  $N_{iao}$  为 1 到 5 的整数. 图 8 和图 9 分别给出了在种群大小不同的情况下跳动和微动数变化对目标函数的影响.

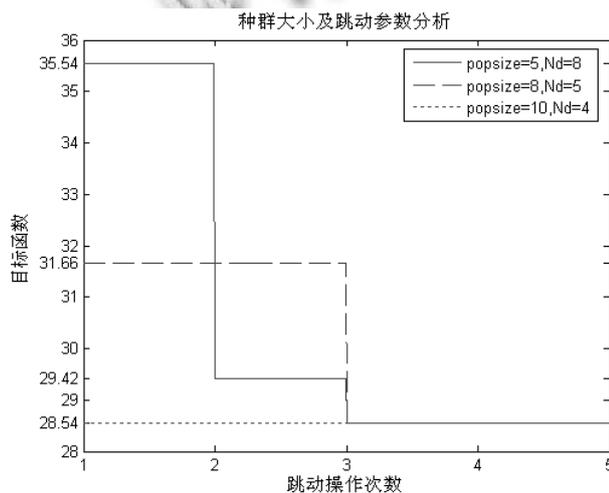


图 8 种群大小及跳动参数分析图

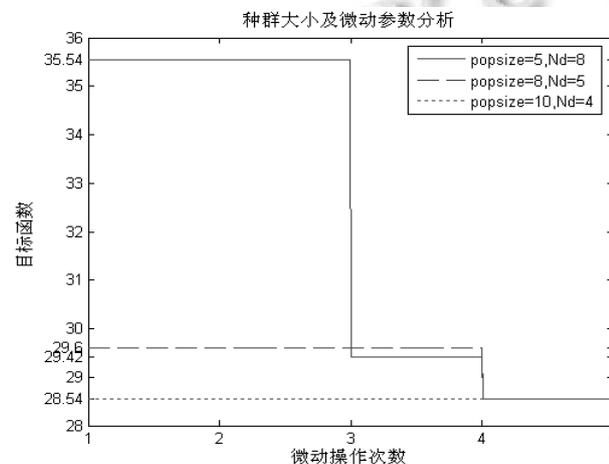


图 9 种群大小及微动参数分析图

从图 8 和图 9 可以看出, 大的种群数量往往会带来较优的初始解, 可行解随微动与跳动操作次数的增多而变得更优. 但是达到一定数目后, 操作次数的增加对解改善程度的影响变小. 所以在应用中, 必须做一些前期试验, 找到合适的跳动与微动数目.

种群数目大, 迭代次数小, 能得到较优的初始解, 但是加大了解的不稳定性, 解的更新速度较慢. 反之, 种群数目小, 迭代次数大, 能使解得的更新速度较快, 但是初始解较差, 难以得到优秀解.

通过数值试验来确定合适的种群数以及迭代数很有必要.

## 7 结论

本文提出了基于拓扑排序的邻域搜索算法, 用项目调度问题库的不同规模的数据进行验证, 结果表明, 该算法都能得到较优的解, 并具有操作简单, 寻优能力强的特点. 本文的研究对多资源约束下项目调度问题的优化具有重要的意义.

### 参考文献

- 1 王宏, 林丹, 李敏强. 一种求解资源受限项目调度问题的自适应遗传算法. 系统工程, 2005, 23(12): 99-102.
- 2 Demeulemeester EL, Herroelen WS. A branch-and-bound procedure for the generalized resource-constrained project scheduling problem. Operations Research, 1997, 45(2): 201-212.
- 3 Patterson J, Roth G. Scheduling a project under multiple resource constraints: A 0-1 programming approach. AIIE Transactions, 1976, 8(4): 449-455.
- 4 敬石开, 苏青, 汪艳兵等. 基于贪心策略的资源约束多项目调度方法. 制造业自动化, 2009, 31(6): 102-105.
- 5 刘金定, 伍艳莲, 叶锡君. 基于拓扑排序资源约束下多项目调度优化算法. 西北大学学报, 2008, 27(6): 49-51.
- 6 廖仁, 陈庆新, 毛宁. 资源约束下多项目调度的启发式算法.

(下转第 41 页)

数据处理层、参数分析层、系统维护层、预测层、决策支持层和表示层可只对一个数据服务器中数据进行分析,即获得单一线路沉降值、变形趋势预测、系统维护性等信息。某一层也可利用多个数据服务器上一层或几层分析结果,进行统一分析,得到上海市所有线路的预测信息,并将其结果保存在中央数据服务器中。例如预测层可通过访问多个数据库服务器中参数分析层、系统维护层信息,预测出整个地铁网络未来变形趋势,以线路交叉点为主要研究对象,提前预警。除此之外,也可由中央服务器一次性从多个数据库服务器中调取原始数据,进行分析及结果展示。

## 5 总结

本文将分布式监测信息系统分为内部系统和外部系统,并分别概括其在结构上的共性,尽可能地降低其模块耦合性,提出了通用的内部系统架构和外部系统架构。对于内部信息系统,采用协调服务器来实现数据库服务器的分布式存储,较好地满足其可靠性及易扩展性。对于外部信息系统,按功能分为独立的模块,使得该系统具有较大的灵活性,能够有效地减弱系统变更所带来的需求变化。基于该架构体系,以上海地铁网为对象,构建了一个分布式地铁隧道沉降变形监测信息系统,该系统可广泛应用于高铁线路监测系统,对于其他领域的分布式监测系统也具有极大的意义。

## 参考文献

- 1 王丰贵.因特网环境下分布式监测系统关键技术研究及应用[学位论文].长春:吉林大学,2009.
- 2 周奇才,陈海燕,熊肖磊等.运营地铁隧道形变检测信息系统设计.中国工程机械学报,2010,8(1):96-100.
- 3 徐俊刚,邵佩英.分布式数据库系统及其应用.北京:科学出版社,2012:91-100.
- 4 Brownjohn J, Tjin SC, Tan GH, Tan BL, Chakraborty S. A structural health monitoring paradigm for civil infrastructure. 1st FIG International Symposium on Engineering Surveys for Construction Works and Structural Engineering. Nottingham, United Kingdom. 2006. 247-256.
- 5 Hsieh YM, Hung YC. A scalable IT infrastructure for automated monitoring systems based on the distributed computing technique using simple object access protocol Web-services. Automation in Construction, 2009, 18(4): 424-433.
- 6 赵炯,鲁丹君,胡玉俊等.地铁隧道沉降变形信息发布及预警系统设计与实现.铁道科学与工程学报,2011,8(6):44-49.
- 7 中华人民共和国国家质量监督检验检疫总局.GB/T 26221-2010 基于状态的维护系统体系结构.http://www.bzfxw.com.
- 8 Cheng MY, Ko CH, Chang CH. Computer-aided DSS for safety monitoring of geotechnical construction. Automation in Construction, 2002, 11: 375-390.
- 9 周奇才,高嵩,熊肖磊等.基于图像传感器网络的隧道形变监测系统研究.计算机测量与控制,2009,17(9):1699-1700.
- 10 Kim S, Pakzad S, Culler D, Demmel J, Fennes G, Glaser S, Turon M. Health monitoring of civil infrastructures using wireless sensor networks. Proc. of the 6th International Conference on Information Processing in Sensor Networks. ACM Press, Cambridge, Massachusetts, USA. 2007. 254-263.
- 11 刘士新,宋健海,唐加福.基于关键链的的资源受限项目调度新方法.自动化学报,2006,32(1):60-66.
- 12 郭研,宁宣熙.利用遗传算法求解多资源项目平衡问题.系统工程理论与实践,2005,17(10):78-82.
- 13 Tsai DM, Chiu HN. Two heuristics for scheduling multiple projects with resources constraints. Construction Management and Economics, 1996, 14: 325-340.
- 14 应瑛,寿涌毅,李敏.资源受限多项目调度的混合遗传算法.浙江大学学报,2009,43(1):23-27.
- 15 刘士新.项目优化调度理论与方法.北京:机械工业出版社,2006:95-100.

(上接第127页)

管理工程学报,2002,16:100-103.

7 Jedrzejowicz P, Ratajczak E. Solving the RCPSP/max Problem by Teams of Agents. Heidelberg: Springer-Verlag Berlin, 2009: 734-743.

8 应瑛,寿涌毅,李敏.资源受限多项目调度的混合遗传算法.浙江大学学报,2009,43(1):23-27.

9 刘士新.项目优化调度理论与方法.北京:机械工业出版社,2006:95-100.

10 刘士新,宋健海,唐加福.基于关键链的的资源受限项目调度新方法.自动化学报,2006,32(1):60-66.

11 郭研,宁宣熙.利用遗传算法求解多资源项目平衡问题.系统工程理论与实践,2005,17(10):78-82.

12 Tsai DM, Chiu HN. Two heuristics for scheduling multiple projects with resources constraints. Construction Management and Economics, 1996, 14: 325-340.