

文章编号:1004-1478(2011)03-0045-05

分布复杂机电系统网络模型图 自动生成算法研究

骆继明, 郭新军

(河南工程学院 电气信息工程系, 河南 郑州 451191)

摘要:为了解决分布复杂机电系统网络模型图的自动生成问题,提出了一种新的基于结构空间的网络图形生成算法:通过定义一个特殊的数据结构用来保存网络模型的基础信息,再按照不同的标准对这些信息进行反复搜索与提取操作,获得的结果分别保存在定义的多个集合中,形成不同的集合分类;根据集合元素的数量以及集合间的关系对集合元素进行迭代计算,不断地辨识每个集合元素的属性和参数,并逐步完成其整个图形绘制.使用网络模型绘制例子对所提出的算法进行了验证,结果表明:此算法具有可行性,能够很好地满足实际需求;同时具有通用性,能够作为共性技术进行推广与应用.

关键词:网络模型图;数据结构;结构空间;迭代计算

中图分类号:TP399

文献标志码:A

Research on auto-generation arithmetic for network model graphs of distributed complex electromechanical systems

LUO Ji-ming, GUO Xin-jun

(Dept. of Electr. Infor. Eng., He'nan Inst. of Eng., Zhengzhou 451191, China)

Abstract: A novel algorithm based on a structure space was presented to realize any automatic network model graph generation for complex systems, in order to solve the problem that the distribution network model of complex electromechanical system automatically generates graphs. A structure space is designed as a special data structure, which is used to save the basic information of the network model graph. Then different sets are defined to manage the classified information in the structure space according to certain operating sequence. The drawn model graph is based on classification sets, which contains the computing and the drawing of graphic objects. In the drawing process, the iterative calculation is used, and the whole graph is displayed with the nodes, relationships, and parameters eventually. Finally, a network model is used to draw an example illustrate the proposed algorithm. It shows that the proposed algorithm is feasible and may satisfy the realistic requirements. Simultaneously, it is discovered that the auto-generation arithmetic for the network model graph has a good universality, and can be widely extended in the application.

Key words: network model graph; data structure; structure space; iterative calculation

收稿日期:2011-04-05

基金项目:河南省科技攻关重点项目(112102210343)

作者简介:骆继明(1975—),男,河南省光山县人,河南工程学院助教,主要研究方向为煤矿综合智能控制技术.

0 引言

分布复杂机电系统网络模型图是用来仿真和分析系统可靠性、安全性等问题的有效工具。它能够让人直观地理解这种复杂问题的演化过程,如故障的传播过程、多因素相互作用、时序变化等。作为一种有效的分析工具和解决问题的手段,网络模型图被广泛地应用于机械、电子、石化等许多行业,并发挥着重要的作用。当前已有的绘图工具很多,但大多是人工绘制图形,效率低。另外,由于人的参与也带来了一些问题,像制作人员必须具备相应的模型制造的专业知识才能完成网络模型的制作过程;由于人的不同知识背景结构和对同一系统不同认识的偏差,制作出来的模型图也不一样等。手工绘制与网络自动生成最大不同是,手工绘制很难反映系统数据的相关性及模型随系统信息的实时变化情况。目前已有的一些关于模型图自动生成技术的研究,或是在通用性方面不够^[1-3]、或是没有形成固定的算法模型^[4]、或是还停留在理论和思想方法阶段^[5],许多生成算法采用单元格定位^[6]然后再进行平滑处理,得到模型网络图。这些方法形成的图形不美观,图形生成过程中大多数元素需要重新定位,计算较为复杂。为此,本文提出基于集合迭运算的网络模型图生成算法,一次性生成网络模型,以满足大数据量元素模型图生成的需要。

1 网络模型分析及处理

分布复杂机电系统网络模型图一般由节点、边、参数、标识等元素组成。节点之间通过边相互关联形成一个相互交织的网络,这些节点和边又以标识相互区别,并使用参数表示元素所具有的定量属性信息。因此,可以使用数学中的图来表示网络模型。假设网络模型记为 G ,模型的节点组成的集合记为 V ,边形成的集合记为 E ,那么,就有如下形式的网络模型关系表示:

$$G = (V, E)$$

$$V = \{v_i | i = 1, \dots, n\} \quad (1)$$

其中, v_i 表示模型的第 i 个节点。

$$E = \{e_i | i = 1, \dots, n\}$$

其中, e_i 表示模型的第 i 条边。

$$P = \{p_i | i = 1, \dots, n\}$$

其中, p_i 表示模型的第 i 个参数。

使用一个数据结构表存储网络模型图的组成要素。对于一个网络模型图而言,只需保存模型的当前节点和当前节点的后继节点,以及相应节点的

编号和该条记录的标号等信息就能够反映整个网络模型的情况。这种表示不会丢失任何基础数据信息,因此是适合的。数据表的结构如下所示。

网络模型表(标识;关联;节点1;节点2;参数)。其形式化表示为

$$R = (id, e_k, v_i, v_c, p_i)$$

为了方便后面的运算,因为 R 主要是反映 v_i 和 v_c 的序列关系,所以将 R 简化表示为一个序偶:

$$R = (v_i, v_c) \quad (2)$$

2 算法实现

2.1 算法设计

网络模型是用来刻画实际系统的,所以网络模型必须反映这一实际,网络模型图也必须能够表达这一客观情况。因此,如果使用有向图网络模型——图中所有的连接都用带有方向的箭头来表示,那么任意2个节点之间不能形成环;对于一般的连通网络,即使用不带箭头的线条直接连接,任意2节点之间只能存在1条连线。这样才符合分布复杂机电系统的实际生产中介质流动的事实,从而使建立的网络模型图满足有用性的要求。

为了方便网络模型的生成,做出如下规定:

1)对节点进行编号,编号必须遵从系统间的统一的耦合顺序的排列。这种排列可以是物质流的流动顺序,也可是电流的流动顺序,或者是系统故障传播顺序,亦或是系统信号的时序发生顺序等。

2)任意2节点之间在数据库中只能保存1条记录,不能发生多个冗余造成自动生成错误。

3)系统模型节点的原始的基础数据必须是完整的,也就是形成网络模型后的整个网络必须是连通的或是弱连通的,不能同时形成多个网,以保证网络模型的唯一性。

2.2 算法描述

使用该算法进行图形绘制时,操作类似于拉网活动,因此也称之为拉网(draw net)算法。模型图自动生成过程如下。

2.2.1 节点的绘制 首先找到具有最小编号的初始节点。如果实际系统是多输入系统,则有多个初始节点,针对这种系统可以通过2种办法去解决:一是虚拟一个初始节点,把现有的多个输入初始节点做为此节点的后继节点;二是在搜索后继节点时,判断是否存在多输入节点,如果存在则直接按照多输入的情况来进行图形绘制处理。接下来搜索与初始节点关联的后继节点,重复搜索直到所有初始节点的后继节点都被搜索完毕为止。对于绘制节点的

定位问题,这里主要考虑美观性、最少线条、最少交叉的原则来分布布局网络的组成要素.

2.2.2 边的绘制 在节点绘制中已经部分地考虑到连线问题,进一步的分析发现任意2个节点的连线可以归结为2种基本形式:不同层次节点的连接如图1a);同层次节点的连接如图1b).所以,边的绘制也就只需处理这2种形式,并且整个网络也就只有这2种形式组成.

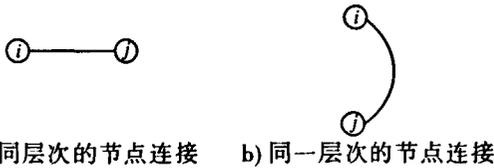


图1 网络模型节点连接的基本形式

2.2.3 属性与参数的标注 要求标注清晰,不产生歧义,同样显示要美观.另外,还要遵从行业的习惯,标注位置的确定依据上述节点和边的位置而定.

2.2.4 各种计算 模型中涉及到多处多种计算问题,其计算方式主要有集合运算、代数运算和逻辑运算.这些计算都是动态计算,计算过程在内存中实施,处理过程只需要基础信息并输出最终计算的结果.处理开始定义了4个结构向量空间A, B, C, D. A保存从数据库中提取的结构数据信息, B保存当前处理节点信息, C保存当前处理节点的暂存信息, D保存当前节点的后继联系信息.首先提取网络模型的所有结构信息,并保存在集合变量A中,接下来进行以下的各种计算.

根据式①和②可知

$$\forall v_i, \forall v_j \in V \Rightarrow R \subseteq V \times V = V^2$$

进一步可知 $A \subseteq V^2, B \subseteq V^2, C \subseteq V^2, D \subseteq V^2$.

1)对提取的数据进行分类计算,假设其结果保存在对应的B, C, D结构向量空间中,则有

$$(A \neq \Phi \wedge B = \Phi) \rightarrow \exists (v_i, v_j)$$

$$(B = \{A(v_i, v_j) \wedge v_i = v_1\})$$

其中 v_1 为初始化编号最小的节点.

$$C = \{(v_i, v_j) \mid \forall (v_i, v_j) \in A(v_i, v_j) \wedge B(v_m, v_n) \wedge v_j = v_m\}$$

$$D = \{C(v_i, v_j) \mid v_i \in V \wedge v_p, v_i \in V \wedge v_p \neq \forall v_i\}$$

2)定位参数的计算,此部分进行的是代数运算.

$$x = x_1 + a \times f \tag{3}$$

其中, x_1 为初始位置; a 为水平均布常量,也就是节点水平跳动步长, $a = l/t$, l 为画布长度, t 为节点水平层次总数; f 为绘制节点的当前水平层次数.式③

保证了节点水平上的均匀定位情况.

$$y = b \times m + (y_1 - b \times (n - 1)) / 2 \tag{4}$$

其中, b 为垂直均布常量,也就是垂直步长, $b = h/n$, h 为画布的宽度, n 为当前绘制节点的垂直层次数; y_1 为初始化垂直方向的位置; m 为当前绘制节点的垂直层次序列数.式④保证了绘制节点垂直方向上沿中心线的对称分布情况.

算法流程见图2.

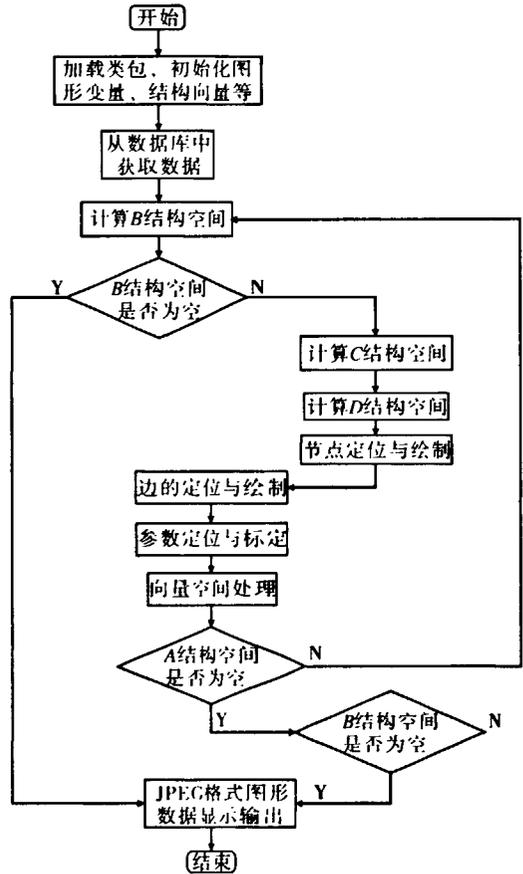


图2 网络模型自动生成算法流程

2.3 算法实例

使用类Java脚本说明算法的实现过程.只给出主要处理方法的程序代码,程序的其他部分文中进行了省略.

1)程序开始,装载Java类包.

```
import iqs. * ; // 加载自定义类库;
import java.awt. * ; // 加载图形显示类库;
import javax.servlet. * ; // 加载Servlet类库;
import com.sun.image.codec.jpeg. * ; // 加载流数据的输出函数库;
```

2)初始化结构向量,原始均为空.

```
private Struct A = new AStruct (); // AStruct ()表示数组结构空间;
```

```
private Struct B = new AStruct ();
```

```
private Struct C = new AStruct ();
private Struct D = new AStruct ();
```

3) 获取数据库中结构向量数据,并保存在结构向量空间 A 中.

```
A = managerLocator.getMAManager()
findByProperties(new String[] { "Aid" },
new Object[] { Mid });
```

4) 初始化图形属性向量:缓冲空间,字体、形状类型、颜色、尺寸等.

```
response.setContentType ("image/gif");
int wid = 1000, hig = 480;
BufferedImage image = new BufferedImage (wid,
hig + 30,
BufferedImage.TYPE_INT_RGB);
Graphics2D G = image.createGraphics ();
private Font mFont = new Font("宋体", Font.PLAIN,
16);
float lineWidth = 2.0f;
((Graphics2D) G). setStroke (new BasicStroke (line-
Width));
```

```
G.setFont (mFont);
```

5) 数据分类,结构保存在不同的类型集合中. 使用 B 空间保持当前处理节点.

```
B.add (A.getAll (vs)); // s 为当前处理节点,初始设
为网络的首节点;
```

使用 C 空间保存 B 空间节点的后继节点. C.add (A.get(B.getAll (ve))); // e 为 B 的后继节点.

6) 使用结构空间 D 保存当前节点的后继关联节点. 绘制节点、边、标注等图形.

利用空间 D 保存 B 的无重复的后继节点. D.add (C.getAll (vue)); // vue 为无重复节点. 定位节点位置.

```
x = 10 + v * (f);
y = h * m + (hig - h * (n_n - 1)) / 2;
绘制节点.
```

```
G.fillOval(x, y, alpha, beta); // alpha, beta 为节点大小参数;
```

定位边的位置,包括起始位置和终点位置. 它又分 2 种情况:同层次节点的边位置确定和不同层次节点的边位置确定.

同层次的边又有由上往下绘制的情况和由下往上绘制的情况.

从上往下的表达式为:

```
x2 = x;
y2 = h * c + (hig - h * (n_n - 1)) / 2;
G.drawArc(x + 15 - 25 * (c - m),
y + 15, 50 * (c - m), h * (c - m), 90,
-180);
```

由下往上的表达式为:

```
G.drawArc(x + 15 - 25 * (m - c),
y2 + 15, 50 * (m - c), h * (m - c), 90, -
180);
G.drawString ("e" + String.valueOf (C.get (n).getPara
()),
(x + x2) / 2 - 15 + 25 * (Math.abs(c -
m)), (y + y2) / 2 + 15);
不同层次边的确定.
x2 = 10 + v * (f + 1);
y2 = h * t + (hig - h * (v_n - 1)) / 2;
G.drawLine(x + 15, y + 15, x2 + 15, y2 + 15);
G.drawString ("e" + String.valueOf (C.get (n).getPara
()),
(x + x2) / 2 + 10, (y + y2) / 2 + 15);
G.drawString ("v" + String.valueOf (B.get (m).get-
Para ()), x + 6, y + 19);
```

7) 如果 A 中数据不空,继续找到 D 集合中节点的后继节点保存到清空的 B 集合中.

```
B.add (A.get (D.getAll (q)));
```

同时,使用 C 集合保存 B 集合节点的后继节点;

使用 D 集合保存无重复的 C 集合节点;

```
C.add (B.getend (e));
```

```
D.add (C.getnoduplication(s));
```

8) 对缓冲区中的图形进行 JPEG 格式的图形编码,使用流方式输出 G 图形,并在 Browser 中显示.

```
G = JPEGImage (OBJ - G);
```

```
G. OUT;
```

至此算法实现完毕.

3 应用实例

此网络模型图的自动生成技术已经完成了实际应用的原型系统的开发过程. 系统采用了 B/S 的框架结构和常用的 Tomcat 5.5 的发布系统. 在这种使用 Browser 浏览器表达图形效果非常困难的情况下,实现了浏览器网络模型图的自动生成与显示工作.

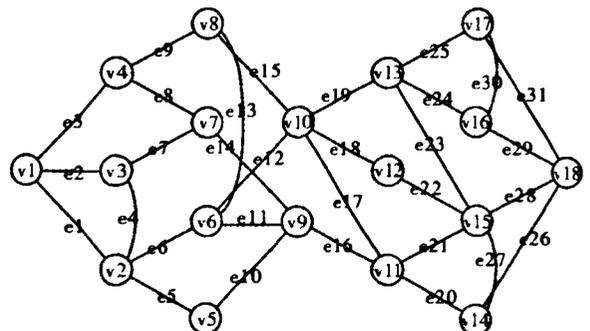


图 3 自动生成算法绘制的网络模型图

图3显示了一个通过使用自动生成算法绘制的网络模型图形.此模型是以国家某个大型工矿企业为原型进行的研究工作,网络模型是由18个节点、31条边及相应的参数和标识组合而成.

4 结论

复杂系统网络模型图自动生成算法能够满足分布式复杂机电系统网络模型生成的需要.采用此种方法绘制图形可以产生手工图形绘制难以达到的效果:布局合理、层次清晰、界面美观、生成速度快、效率高,能够实现复杂系统故障分析的直观表示,使复杂问题变得简洁;实现了系统内在外在呈现的一致表达,动态地反映了实际分布复杂系统内部关联耦合的运动变化情况,避免了因表达困难而带来的危险性.本研究解决了分布复杂机电系统安全的一个重要的网络自动生成问题,更加丰富的表达形式和更加智能的系统图形安全控制方法还有待进一步研究,系统安全问题的最终解决,需要与更多的决策控制方法相结合才能实现.

(上接第7页)

表1 重建误差匹配数据1

图像	大小	搜索范围	重建阶数	重建误差 <i>err</i>		匹配率/%
				左图像	右图像	
Tsukuba	384 × 288	16	60	8.8	8.8	65.2
Map	284 × 216	30	45	16.7	15.9	76.8
Venus	434 × 383	20	50	9.2	9.3	70.7
Swtooth	434 × 380	20	50	9.6	9.5	69.6

表2 重建误差匹配数据2

图像	大小	搜索范围	重建阶数	重建误差 <i>err</i>		匹配率/%
				左图像	右图像	
Tsukuba	384 × 288	16	10	18.0	17.8	68.5
Map	284 × 216	30	10	24.0	23.3	77.9
Venus	434 × 383	20	10	19.1	19.2	73.3
Swtooth	434 × 380	20	10	18.7	19.2	73.5

了立体匹配工作的研究.针对不同阶的重建误差图像,得到的匹配结果也不尽相同.实验结果表明,图像重建的误差信息也可以作为一种匹配度量来进

参考文献:

- [1] Vempati Chaitanya, Campbell Matthew. A graph grammar approach to generate neural network topologies [C]//2007 Proc of the ASME Int Design Eng Tech Conf and Comp and Infor in Eng Conf, [s. l.]: ASME, 2007: 79-89.
- [2] 雷蕾,郭林,纪越峰.一种应用于不对称网络中的生成树拓扑抽象算法[J].电子与信息学报,2006,28(10):1917.
- [3] 田永青,杜国宁,李志,等.基于云理论神经网络决策树的生成算法[J].上海交通大学学报,2003,37(5):113.
- [4] 张彩庆,王婷.网络图自动生成算法研究[J].数学的实践与认识,2003,33(12):9.
- [5] 刘学军,贾亚洲,张日明.数控机床可靠性智能网络系统控制模型及自动生成研究[J].机械工程学报,2003,39(9):114.
- [6] 徐斌,钱德沛,陆月明,等.一种基于抽象点的网络拓扑自动生成算法[J].小型微型计算机系统,2001,22(4):411.

行匹配工作.

参考文献:

- [1] Flusser J. Pattern recognition by affine moment invariants [J]. Pattern Recognition, 1993, 26(1):167.
- [2] Heywood M I. Fractional central moment method for movement invariant object classification [J]. IEEE proceedings of Vision Image and Signal Processing, 1995, 142(4):213.
- [3] Mukundan R, Ong S H, Lee P A. Image analysis by Tchebichef moments [J]. IEEE Trans on Image Proc, 2001, 10(9):1357.
- [4] Mukundan R, Ong S H, Lee P A. Continuous Orthogonal Moments for Image Analysis [C]//Int Conf of Imaging Sci Syst and Tech, New Zealand: University of Canterbury Press, 2001:23-29.
- [5] 夏永泉,刘正东,杨静宇.一种基于正交矩的立体匹配方法[J].系统仿真学报,2005,17(9):2082.