

基于斥力-张力模型的网络拓扑图布局算法

程远, 严伟, 李晓明

(北京大学计算机科学技术系网络与分布式系统实验室, 北京 100871)

摘要: 针对网络管理系统中网络拓扑图的自动布局问题, 提出了一种新的算法。赋予拓扑图的顶点和边不同的物理性质(质量、张力、斥力等), 各个顶点在力的相互作用下不断运动, 直至最终达到力的平衡, 同时完成了拓扑图的自动布局。

关键词: 网络管理; 网络拓扑图; 自动布局; 斥力-张力模型

Network Topology Autolayout Algorithm Based on Repulsion-Tension Model

CHENG Yuan, YAN Wei, LI Xiaoming

(Computer Networks and Distributed Systems Lab, Department of Computer Science and Technology, Peking University, Beijing 100871)

[Abstract] A new algorithm is advanced in this paper to do the autolayout of the network topology. All the vertexes and edges of a graph are assigned to some physical properties such as mass, repulsion and tension. Thus these vertexes will keep moving along the direction of composition of forces until the system comes to the state of balance when the layout of the graph has done automatically.

[Key words] Network management; Network topology; Autolayout; Repulsion-Tension model

1 概述

网络管理系统大多是以网络拓扑图(又称网络地图)为操作核心的,即以图形化的方式显示子网、路由器等网络设备的逻辑连接关系和其它相关信息,操作人员可以在网络拓扑图中直接进行配置、性能、故障、计费、安全等管理操作。网络拓扑图的构造涉及到两个问题:网络设备的自动发现和拓扑图的自动布局。当前对于网络设备的自动发现已经有了较为深入的研究,算法也基本成熟^[1,2];而拓扑图的自动布局算法方面的研究工作相对较少,很多商业网络管理软件的自动布局功能也比较简单^[3,4]。

目前网络拓扑图的自动布局算法主要有如下几种:(1)行列布局,即简单地把图中所有顶点等距排列然后进行连线^[5];(2)环形布局,按照同心圆的方式布局顶点再连线^[6];(3)引力布局^[5],引入了力的概念,但收敛速度慢,支持顶点少。以上算法往往需要按照某些原则一次性地把每一个顶点的坐标计算好,但是对于比较复杂的拓扑图(顶点数大于20,有环路存在)来说,这些算法得出的结果往往会显得杂乱无章,需要较多的人工干预。

本文提出的自动布局算法期望达到如下的效果:尽量少的边交叉,顶点分布距离均匀,有直接连线的顶点尽可能分布在一处。

2 自动布局算法

2.1 算法目标

本算法的目标是得到一个“好”的拓扑图。本文提出如下的二元组作为评价自动布局结果优劣的参数。

对于一个布局完成的拓扑图 $G(V, E)$:其中 V 为 n 个顶点的集合 $\{v_1, v_2, v_3, \dots, v_n\}$,并且每一个顶点 v_i 有一个已经计算出来的坐标 (x_i, y_i) ; E 为 m 条边的集合 $\{e_1, e_2, e_3, \dots, e_m\}$,则可以计算出每一条边 e_i 的长度。

定义评价参数 $S(G)=(\alpha, \beta)$,其中: α 定义为图 G 中边交叉点的个数。显然 α 越小,图的边交叉越少,布局越清

晰。 β 定义为图 G 中所有边的长度的平均绝对偏差和平均值的比值,即

$$\beta = \frac{\sum |e_i - \bar{e}|}{\bar{e}}$$

其中 \bar{e} 为所有边长度的算术平均值。 β 代表了顶点分布的均匀程度, β 越小,图中顶点分布越均匀。

参数 $S(G)$ 的比较规定为:首先比较 α , α 的大小决定 $S(G)$ 的大小;若 α 相等,则比较 β ,此时 β 的大小决定 $S(G)$ 的大小。 $S(G)$ 越小,拓扑图 G 的布局越好。

2.2 算法思想

从最简单的拓扑形式入手,考虑如图1的3种拓扑结构:

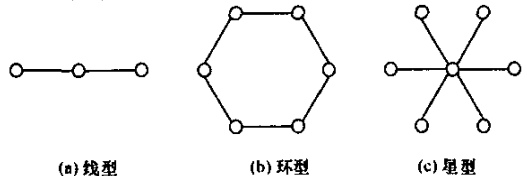


图1 3种拓扑结构

上图依次为线型、环型和星型的简单拓扑,当我们手工为其布局时,比较自然的形式就如图1所示。可是进一步深入考虑的话,会发现这3种形式背后有某种共同的东西:对称与均匀。各个顶点保持距离是由于斥力的作用,由于边的联系彼此之间又没有完全散开。

受此启发,考虑如下的情景:对于任意的拓扑图来说,把每一个顶点看作一个质点,它有自己的质量;把每一条边看作一条橡皮筋,它有自然长度(即不受外力时的长度),如果被拉长,则表现出张力;任意两个顶点之间互相有斥力的作用。这样的一个物理系统,定义了张力和斥力的公式以

作者简介:程远(1978—),男,硕士生,研究方向为计算机网络;严伟,副教授;李晓明,教授、博导

收稿日期:2003-02-11

E-mail:chengyuan@net.cs.pku.edu.cn

后, 给图中每一个质点(顶点)一个随机的位置, 就可以让其按照物理规律自行运行; 在每一步中, 计算各个顶点受到的合力, 并让该顶点沿合力方向位移一定的距离, 直至最终达到平衡。

2.3 算法描述

对于任意的无向图 $G(V, E)$, 其中 V 为 n 个顶点的集合 $\{v_1, v_2, v_3, \dots, v_n\}$, E 为 m 条边的集合 $\{e_1, e_2, e_3, \dots, e_m\}$ 。定义如下:

- 顶点 v_i 的度数为 $Degree(v_i)$;
- 顶点 v_i 的质量为 $Mass(v_i)$;
- 边 e_i 的自然长度为 $Length(e_i)$;
- 顶点 v_i, v_j 的距离为 $Distance(v_i, v_j)$;
- 任意两个顶点 v_i, v_j 之间有斥力, 定义 v_i 所受斥力为 $Repulsion(v_i, v_j)$, v_j 所受斥力为 $Repulsion(v_j, v_i)$;
- 相邻的两个顶点 v_i, v_j 受它们之间的边 e_i 的张力, 定义 v_i 所受张力为 $Tension(v_i, v_j, e_i)$, v_j 所受张力为 $Tension(v_j, v_i, e_i)$;
- 顶点 v_i 所受的合力为 $Force(v_i)$ 。

上面的张力、斥力以及合力都是矢量, 下面的运算也都是矢量的加法, 具体实现时可以正交分解为 X 轴和 Y 轴的力, 符号“ \propto ”代表成正比。

输入: 图 $G(V, E)$

for V 中的每一个顶点 v_i

在画布的范围随机为 v_i 分配坐标 (x_i, y_i)

$Force(v_i) = 0$;

end for

while 系统没有平衡 do

for E 中的每一条边 e_i

找出其对应的两个顶点 v_i, v_j ;

$Force(v_i) += Tension(v_i, v_j, e_i)$;

$Force(v_j) += Tension(v_j, v_i, e_i)$;

end for

for V 中的每一个顶点 v_i

for V 中每一个顶点 v_j

if $(v_i \neq v_j)$

$Force(v_i) += Repulsion(v_i, v_j)$;

end if

end for

end for

for V 中的每一个顶点 v_i

沿 $Force(v_i)$ 的方向移动距离 d ,

$$|d| \propto \frac{|Force(v_i)|}{Mass(v_i)} \quad (1)$$

表现为 (x_i, y_i) 变化;

$Force(v_i) = 0$;

end for

end while

输出: V 中每一个顶点 v_i 的坐标 (x_i, y_i) 。

2.4 算法讨论

(1) 算法的性质

在理想条件下, 对于这样一个物理系统, 所有顶点的初始速度为0, 即系统的整体初动量为0, 而系统又不受外力的作用, 从而动量守恒, 系统的总动量一直为0。这就保证了系统的最终阶段要么内部顶点振荡, 要么静止, 而不可能整体向某一方向移动或旋转。

具体实现的时候, 在式(1)的地方, 可以让 $|d|$ 不超过一个常量, 比如8个像素。这样每一步每个顶点最多移动8个像素。振荡的状态可以监测出来, 发生振荡时, 可以适当减少每一步的移动量, 直至静止。系统振荡或静止时算法就可以中止。

根据物理意义, 算法的结果满足: 有边(“橡皮筋”)连接的顶点会分布在一处, 距离相近的顶点分布均匀(斥力作用), 边的交叉尽可能少(斥力作用)。

(2) 物理公式的选取及其影响

1) 张力公式: 张力和边的自然长度 $Length(e_i)$ 以及边的弹性系数 k 有关, 公式满足胡克定律:

$$|Tension(v_i, v_j, e_i)| = \begin{cases} 0, & (Length(e_i) \geq Distance(v_i, v_j)) \\ k * (Distance(v_i, v_j) - Length(e_i)), & (Length(e_i) < Distance(v_i, v_j)) \end{cases}$$

实现者需要决定 k 的大小和每一条边的自然长度的大小。 k 越大, $Length$ 越小, 则布局越密集。 $Length$ 的选取要考虑到画布的大小和顶点的多少问题。

2) 斥力公式: 斥力存在于任意两顶点之间, 和顶点的质量成正比, 和距离的平方成反比。公式如下:

$$|Repulsion(v_i, v_j)| = \begin{cases} f, & (Distance(v_i, v_j) = 0) \\ \frac{g * Mass(v_i) * Mass(v_j)}{Distance(v_i, v_j)^2}, & (Distance(v_i, v_j) \neq 0) \end{cases}$$

注意当两顶点重合时, 定义其斥力为一个常量 f , 方向随机。实现者需要决定 g 的大小, g 越大, 布局越稀疏, 越快趋向平衡状态。

3) 质量公式: 可以考虑质量和顶点的度数成正比, 对于网络拓扑中常见的星型结构, 中心顶点的度数越大, 其质量越大, 则斥力越大, 其周围顶点就有更多的空间分布。

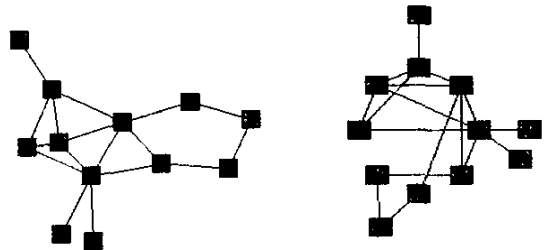
根据上面的讨论, 下文应用实例中的参数取值如下:

$$Length(e_i) = DefaultLength(\text{建议取值} 50), k = 1, g = DefaultLength^2, Mass(v_i) = Degree(v_i)。$$

3 应用实例

3.1 本算法和HP Openview NNM 6.1自动布局算法的比较

图2为对同一个网络拓扑图应用不同自动布局算法的结果。图2(a)为本算法自动布局的结果, $S(G_{left}) = (0, 0.13)$; 图2(b)为商业网管软件HP Openview NNM6.1自动布局的结果, $S(G_{right}) = (7, 0.32)$ 。可以看出: 本算法的结果没有边互相交叉, 并且顶点之间的相连关系表现得十分明晰; 而右图有多处边交叉, 显得比较杂乱。评价参数的计算结果也是 $S(G_{left}) < S(G_{right})$ 。



(a) 应用本算法的结果 (b) 应用 HP Openview NNM6.1的结果

图2 应用不同自动布局算法的结果

3.2 北京大学校园网网络拓扑(部分)

图3为北京大学校园网(部分)的网络拓扑信息利用本算法自动布局后形成的网络拓扑图。图中白色顶点代表路由器, 深色顶点代表子网。可以看出自动布局的结果较好地符合了算法的预期结果: 和路由器直接相连的子网呈星型均匀

(下转第188页)

```

str1 = str1 & "@" & Dt.DataSourceAlias
str2 = objLayers(II).Name
ListView1.ListItems.Add , str2, str1
Next II
Set objLayers = Nothing
Set objGeopoint = Nothing
End Sub
//ListView1 和 ListView2 的联动
Private Sub ListView1_ItemCheck(ByVal Item As MSCOMCTL.IB.
ListIteM)
Dim objGeopoint As New soGeoPoint
Dim objGm As soGeometry
Dim objGeocircle As New soGeoCircle
Dim objGC As New soGeoCompound
Dim objRt As soRecordset
Dim strFieldName As String '要显示字段的名称
Dim strAFieldName(1 To 25) As String
Dim strAFieldName1(1 To 3) As String
Dim strAFieldName2(1 To 30) As String
Dim strLN as sTring
Dim objListItem As ListIteM
Dim objStyle As New soStyle
Deleteheaders
Dim ww As Integer, aa As Integer
strLN = Item.Key
Set objDv = frmMain.ActiveForm.SuperMap.Layers(strLN).dataset
For aa = 3 To objDv.FieldCount
Dim objFi As soFieldInfo
Set objFi = objDv.GetFieldInfo(aa)
Dim KK As Integer
KK = aa - 2
strAFieldName(KK) = objFi.Name
Dim strKeyvalue As String
strKeyvalue = Str(aa) & "ww"
ListView2.ColumnHeaders.Add aa, strKeyvalue, strAFieldName
(KK)
Next aa
ListView2.ListItems.Clear
objGeopoint.x = pointX
objGeopoint.y = pointY
Set objRt = objDv.QueryByDistance(objGeopoint, dTolerance, "")
'获取记录

```

(上接第105页)
分布在路由器的周围,各个路由器之间的互联关系一目了然。计算 $S(G)=(1,0.19)$ 。该图的拓扑信息来自北京大学计算机系网络实验室开发的网络管理系统CSManager^[6]对于北大校园网的自动发现。

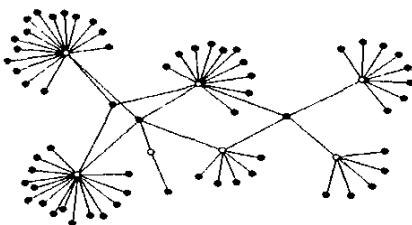


图3 北京大学校园网的网络拓扑图

```

objRt.MoveFirst
Do Until objRt.IsEOF
'逐个算距离
Set objGm = objRt.GetGeometry
If Not objGm Is Nothing Then
'放到ListView2中
Set objListItem = ListView2.ListItems.Add(,"ID" & objRt.GetID)
objListItem.SubItems(1) = Str$(objGeopoint.DistanceTo(objGm))
Dim ffname As Integer
For ffname = 2 To objDv.FieldCount
If strAFieldName(ffname - 1) <> "" Then
objListItem.SubItems(ffname) = objRt.GetFieldValue(strAField
Name(ffname - 1)) & vbNullString
'添加图像到objGC
End If
Next
objGC.AddPart objGm
End If
objRt.MoveNext
Loop
End sub

```

3 结束语

本项目经多方调研,考虑到价格、功能、技术支持等多方面的因素,选取北京超图地理信息技术有限公司的组件式GIS,因为SuperMap具有无缝集成特性、开发语言的多样性、良好的扩展性、强大的伸缩性和流行的可视化程序设计且价格便宜、技术支持及时。作者成功地开发出了森林防火信息系统,火场周边信息的自动提取技术是本系统的一个代表模块,该模块功能得到了用户的好评,将准备进一步改进,采用数据挖掘技术,达到信息智能提取的水平。

参考文献

- 1 邸雪颖,王宏良.林火预测预报.哈尔滨:东北林业大学出版社,1994
- 2 臧淑英,万鲁河.地理信息系统基本原理及软件操作.哈尔滨:地图出版社,2000-06
- 3 周炎坤,李满春.大型空间数据仓库初探.测绘通报,2000,(8):22-23
- 4 宋关福,钟耳顺.组件式地理信息系统研究与开发.中国图像图形学报,1998,3(4):313-317

4 结论

本算法为网络拓扑图的自动布局提供了一种新的思路,并且提出了一个评价自动布局结果优劣的参数。实验证明本算法是简洁有效的。

参考文献

- 1 徐斌,王磊.网络拓扑信息提取的研究.微型机与应用,2000,(4):34-35
- 2 凌军,曹阳.基于ARP和SNMP的网络拓扑自动发现算法.武汉大学学报(理学版),2001,47(1):67-70
- 3 AdventNet Web NMS 2.3. <http://www.adventnet.com.cn/index.html>
- 4 HP Openview NNM 6.1. <http://www.openview.hp.com/products/nnm/index.asp>
- 5 j2sdk1.4.0 Demo GraphLayout.<http://java.sun.com/j2se/1.4/docs/relnotes/demos.html>
- 6 北京大学计算机系网络实验室.综合网络管理系统CSManager. <http://net.cs.pku.edu.cn/~network/>