[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200580028110.8

[43] 公开日 2007 年 7 月 25 日

[11] 公开号 CN 101006443A

[22] 申请日 2005.8.10
[21] 申请号 200580028110.8
[30] 优先权
[32] 2004. 8.16 [33] US [31] 10/918,713
[86] 国际申请 PCT/US2005/028521 2005.8.10
[87] 国际公布 WO2006/023357 英 2006.3.2
[85] 进入国家阶段日期 2007.2.16
[71] 申请人 特里诺尔公司
地址 挪威福尔内伯
[72] 发明人 吉奥夫雷·坎赖特
肯斯·恩格－蒙森 马克·布尔格斯
[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 李春晖

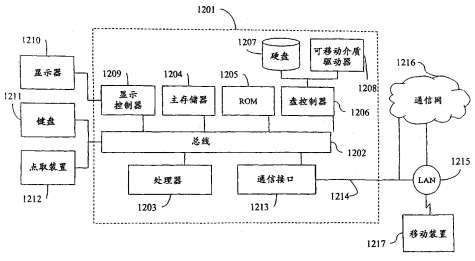
权利要求书 6 页 说明书 25 页 附图 9 页

[54] 发明名称

具有陷补救的用于使用链接分析进行文档排序的方法、系统和计算机程序产品

[57] 摘要

一种具有陷补救的、用于使用链接分析对文档排序的方法、设备和计算机程序产品，包括：从包含链接和节点的原始图形成元图；以及如下两个步骤之一：反向元图中的链接和泵压元图中的源。



1. 一种在计算机系统中使用链接分析来排序文档的方法，包括:

在每个节点代表一个文档并且每个链接代表一个文档中对一个其它文档的引用的原始图中，识别所有的强连接的组件(SCC);

通过用元节点替代每个SCC来形成元图;

以下两个步骤之一

通过为元图中每一对已链接的SCC在所述原始图中增加至少一个链接来修改所述原始图，以使该对已链接的SCC变成强连接的，和

泵压与元图中的源相对应的一个或多个SCC;

确定链接分析节点权重；以及

当确定所述文档的排序时使用所述链接分析节点权重。

2. 根据权利要求1的方法，其中：形成元图的步骤包括：，

保持第一和第二SCC之间的链接，以使从第一SCC中的节点到第二SCC中的节点的有向链接变成从第一元节点到第二元节点的链接。

3. 根据权利要求2的方法，还包括:

识别对应于所述元图的挤压图。

4. 根据权利要求1的方法，其中，在所述原始图中增加至少一个链接的步骤包括:

识别包括开始点和结束点在内的SCC间链接；以及

为所述SCC间链接增加反向链接。

5. 根据权利要求4的方法，还包括:

分配权重ε给所述增加的反向链接。

6. 根据权利要求5的方法，还包括:

分配一个比对应的SCC间链接的计算出的权重更低的权重ε给所述增加的反向链接。

7. 根据权利要求1的方法，还包括:

确定文本相关性节点权重；以及

当确定所述文档的排序时，与所述链接分析节点权重一起使用所述文本相关性节点权重。

8. 根据权利要求1的方法，其中，所述确定链接分析节点权重的步骤包括:

为所述SCC确定邻接矩阵，所述确定邻接矩阵的步骤应用正向和反向运算符之一，其中，所述正向和反向运算符之一是标准化运算符和非标准化运算符中的一个。

9. 根据权利要求2的方法，其中，泵压一个或多个SCC的步骤还包括:

使用挤压图来确定在挤压图中只有出链接的一个或多个元节点；以及

选择与所述一个或多个元节点相对应的SCC作为所述一个或多个SCC。

10. 根据权利要求1的方法，其中，泵压一个或多个SCC的步骤还包括:

识别所述一个或多个SCC内的每个SCC内链接;

为所述一个或多个SCC的每一个确定邻接矩阵，所述确定邻接矩阵的步骤包括：应用正向和反向运算符之一，其中，所述正向和反向运算符之一是标准化运算符和非标准化运算符中的一个;

通过基于所述邻接矩阵计算所述一个或多个SCC中的每一个的主特征值来为所述一个或多个SCC的每一个确定增益；以及

增大所述一个或多个SCC的每一个的增益，直到满足如下三个条件:

(i) 所述一个或多个SCC的每一个具有相同的增益;

(ii) 所述一个或多个SCC的每一个的公共增益大于任何非源SCC的增益；和

(iii) 所述一个或多个SCC的公共增益大于任何增益未被如此增大的源SCC的增益。

11. 根据权利要求10的方法，还包括:

确定所述一个或多个SCC中的任一个是否包括单个节点并且没有SCC内链接；以及

在识别每个SCC内链接的步骤之前，对于被确定为包括单个节点并且没有SCC内链接的任何一个或多个SCC，增加从所述单个节点指向它本身的自链接。

12. 根据权利要求10的方法，其中，增大所述一个或多个SCC的每一个的增益的步骤包括：

把SCC中的全部原始的SCC内链接乘以因子G/g，其中，G表示一个或多个SCC的每一个的期望公共增益，而g表示该SCC的原始增益。

13. 根据权利要求1的方法，还包括：

为指向节点和被指向节点中的至少一个计算个人兴趣分数；以及

根据所述个人兴趣分数，调整从所述指向节点向外指出或者指向所述被指向节点的至少一个链接的权重。

14. 根据权利要求1的方法，还包括：

为至少一个节点计算个人兴趣分数；

把计算的个人兴趣分数汇集成个性化补充向量；以及

在节点权重计算过程的每次迭代处，将所述个性化补充向量加到计算出的节点权重向量，以确定所述链接分析节点权重。

15. 根据权利要求1的方法，还包括：

为至少一个节点计算个人兴趣分数；和

把该个人兴趣分数加到已确定的链接分析节点权重。

16. 根据权利要求1的方法，其中，借助于收集有关所述文档的信息的爬行器来识别所述原始图。

17. 根据权利要求16的方法，其中，所述爬行器是web爬行器，并且所述文档是环球网上的网页。

18.一种被配置来使用链接分析排序文档的计算机系统，包括：

存储指令的存储器和执行所述指令以便完成如下步骤的处理器：

在每个节点代表一个文档并且每个链接代表一个文档中对一个

其它文档的引用的原始图中，识别所有的强连接组件(SCC);

通过用元节点替代每个SCC来形成元图;

以下两个步骤之一

通过为元图中的每对已链接SCC在所述原始图中增加至少一个链接来修改原始图，以使该对已链接的SCC变为强连接的，和

泵压与元图中的源相对应的一个或多个SCC;

确定链接分析节点权重；以及

当确定所述文档的排序时，使用所述链接分析节点权重。

19. 根据权利要求18的系统，其中：形成元图的步骤包括:

保持第一和第二SCC之间的链接，以使从第一SCC中的节点到第二SCC中的节点的有向链接变成从第一元节点到第二元节点的链接。

20.根据权利要求19的系统，还被配置来完成如下步骤:

识别对应于所述元图的挤压图。

21.根据权利要求18的系统，其中，在所述原始图中增加至少一个链接的步骤包括:

识别包括开始点和结束点在内的SCC间链接；以及

为所述SCC间链接增加反向链接。

22.根据权利要求21的系统，还被配置来完成如下步骤:

分配权重ε给所述增加的反向链接。

23. 根据权利要求22的系统，还被配置来完成如下步骤:

分配一个比对应SCC间链接的计算出的权重更低的权重ε给所述增加的反向链接。

24. 根据权利要求18的系统，还被配置来完成如下步骤:

确定文本相关性节点权重；以及

当确定所述文档的排序时，与所述链接分析节点权重一起使用所述文本相关性节点权重。

25.根据权利要求18的系统，其中，所述确定链接分析节点权重的步骤包括:

为所述SCC确定邻接矩阵，所述确定邻接矩阵的步骤应用正向和反向运算符之一，其中所述正向和反向运算符之一是标准化运算符和非标准化运算符中的一个。

26. 根据权利要求19的系统，其中，泵压一个或多个SCC的步骤还包括：

使用挤压图来确定在挤压图中只有出链接的一个或多个元节点；和

选择与所述一个或多个元节点相对应的SCC作为所述一个或多个SCC。

27.根据权利要求18的系统，其中，泵压一个或多个SCC的步骤还包括：

识别所述一个或多个SCC内的每个SCC内链接；

为所述一个或多个SCC的每一个确定邻接矩阵，所述确定邻接矩阵的步骤包括：应用正向和反向运算符之一，其中，所述正向和反向运算符之一是标准化运算符和非标准化运算符中的一个；

通过基于所述邻接矩阵计算所述一个或多个SCC中的每一个的主特征值来为所述一个或多个SCC的每一个确定增益；以及

增大所述一个或多个SCC的每一个的增益，直到满足如下三个条件：

(i) 所述一个或多个SCC的每一个具有相同的增益；

(ii) 所述一个或多个SCC的每一个的公共增益大于任何非源SCC的增益；和

(iii) 所述一个或多个SCC的公共增益大于任何增益未被如此增大的源SCC的增益。

28.根据权利要求27的系统，还被配置来完成如下步骤：

确定所述一个或多个SCC中的任一个是否包括单个节点并且没有SCC内链接；以及

在识别每个SCC内链接的步骤之前，对于被确定为包括单个节点并且没有SCC内链接的任何一个或多个SCC，增加从所述单个节点指

向它本身的自链接。

29.根据权利要求27的系统，其中，增大所述一个或多个SCC的每一个的增益的步骤包括:

把SCC中的全部原始的SCC内链接乘以因子G/g，其中，G表示一个或多个SCC的每一个的期望公共增益，而g表示该SCC的原始增益。

30.根据权利要求18的系统，还被配置来完成如下步骤:

为指向节点和被指向节点中的至少一个计算个人兴趣分数；以及

根据所述个人兴趣分数，调整从所述指向节点向外指出或者指向所述被指向节点的至少一个链接的权重。

31.根据权利要求18的方法，还被配置来完成如下步骤:

为至少一个节点计算个人兴趣分数;

把计算的个人兴趣分数汇集成个性化补充向量；以及

在节点权重计算过程的每次迭代处，将所述个性化补充向量加到计算出的节点权重向量，以确定所述链接分析节点权重。

32.根据权利要求18的系统，还被配置来完成如下步骤:

为至少一个节点计算个人兴趣分数；和

把该个人兴趣分数加到已确定的链接分析节点权重。

33.根据权利要求18的系统，还包括爬行器，它能够收集有关所述文档的信息并构建所述原始图。

34.根据权利要求33的系统，其中，所述爬行器是web爬行器并且所述文档是环球网上的网页。

35. 一种计算机程序产品，包括指令，所述指令被配置为使计算设备可执行如权利要求1-17之一所述的步骤。

具有陷补救的用于使用链接分析进行文档排序的方法、系统和计算机程序产品

相关申请的交叉引用

本申请要求2004年8月16日提交的美国申请序列号10/918,713的优先权，其全部内容在此通过参考被合并。本申请还包含与2003年10月29日提交的美国专利申请序列号10/687,602相关的主题，其全部内容在此通过参考被合并。

技术领域

本发明包括一种利用超文本链接对在分布式网络中找到的信息源进行排序的方法、系统和计算机程序产品。具体地说，本发明涉及对来自分布式网络环境中的搜索的命中进行基于链接分析的排序。本方法的软件/固件构成了借助对来自分布式网络环境中的搜索的命中进行基于链接分析的排序，用于搜索分布式信息系统的一个系统的一个组件。该方法适用于文档或其它文件通过链接相关的环境，例如因特网。

背景技术

图1是因特网的基本表示，示出了共同用来构造环球网(WWW)的搜索引擎的多个部分。爬行器1收集关于出现在WWW 2上的网页的信息。所有相关文本信息被馈送到倒排索引3中，用作在WWW的爬过部分上的可用信息的脱机快照。关于链接结构的信息——即每个网页正指向哪些其它网页——被保存在一个链接数据库4中。当用户执行搜索时，她发出一个搜索查询5，此查询被发送给倒排索引3。扫描倒排索引的结果是一个未区分优先级的命中列表。这个命中列表然后根据文本相关性6和链接结构7而被排序。两个排序措施然后整合成一个有

优先次序并经过排序的列表8，此列表8作为一个有优先次序的搜索结果9被返回给始发该搜索查询的用户。

当从倒排索引中获得查询结果时，它们一般将包含位于因特网上的不同WWW域上的命中/文档。文档相互引用(指向)的方式暗中构成一个有向图。这个有向图由作为节点的文档和作为有向边的超文本链接组成，正是这个有向图被用于基于链接的排序。基于链接的排序然后不但基于命中(文档)自身的内容而且还基于它们如何定位在更大的信息网络(有向图)中来估计这些命中(文档)的"权重"或"重要性"。

基于链接分析的排序在以下环境中是有用的：其中，待排序的文档通过从一个文档指向另一文档的定向链接建立关系，并且其中，链接可以被理解为一种推荐的形式。即，从文档*u*指向文档*v*的一个链接意味着对文档*u*感兴趣的用户还可能对文档*v*感兴趣。链接分析允许人们按照一种有用的方式合并包含在所有这些'推荐'(链接)中的信息，因此人们可以在全局上对文档排序。这种方法的突出例子是Google的PageRank方法对被称为环球网的链接文档集合的应用。

这里存在着一些执行基于链接的排序并寻找文档'权重'的替换方式。在不同修正方案中的所有方法都基于查找图形的邻接矩阵*A*的主特征向量(与最大特征值相关的特征向量)。Google的PageRank方法(下面讨论)通过计算列被归一化的转置邻接矩阵的主特征向量来获得每个文档的排序。在HITS方法中，由于Kleinberg(下面讨论)，获得两个排序：1) 通过计算用其自身的转置矩阵构成的邻接矩阵的主特征向量来获得中心分数(hub score)；和2)通过获得用它本身的邻接矩阵构成的转置邻接矩阵的主特征向量来计算出权限分数(authority score)。然而，没有任何实施方案解决因未修改的邻接矩阵(单独被使用)及其转置(同样单独被使用)而来的排序。

通过定义两个简单的运算符-*F*(正向)和*B*(反向)-以及它们各自的标准化形式*f*和*b*，则最容易解释链接分析的各种方法。按照随机漫游的精神，可想象与有向图上的每个节点相关的某一权重(一个正数)。*F*运算符在每个节点*u*处采用权重w(*u*)并正向发送它，即，发送到被节点

*u*指向的所有节点。*B*运算符与箭头相反地发送w(*u*)，即，发送到指向节点*u*的每个节点。*B*是邻接矩阵*A*，而*F*是它的转置。*f*是*F*的列标准化形式；它在节点*u*处采用权重w(*u*)，用节点*u*的输出端数（outdegree）$k_{out,u}$除以它，并发送结果$w(u)/k_{out,u}$给被节点*u*指向的每个节点。类似地，*b*是反向运算符*B*的标准化形式。

PageRank使用被"随机冲浪"运算符(参见下面)补充的*f*(标准化的正向)运算符。HITS方法使用复合运算符*FB*来获得权限分数，并且使用*BF*来获得中心分数。本发明可以与遭受陷问题的任何运算符一起使用，尤其是基本运算符*F*、*B*、*f*或*b*中的任何一个。

所有基于链接的排序方案必须处理的一个问题是在有向链接图结构中的'陷(sink)'情况。一个'陷'是一个节点或者一组节点，它只具有指向它的链接，但没有从该组陷节点指向位于该组陷节点之外的其它节点的链接。典型情况下，陷由一组节点而非一个节点组成；这样一个组被称为一个'陷区域'。同时，一个陷区域中的每个节点都被称为'陷节点'。

在有向图上随机漫游的一个问题是：它们很容易陷入到图形的陷区域中，有方法进入却无法出来。PageRank通过以某种概率加入完全随机的跳跃(与链接无关)来纠正陷，而WiseNut通过采用一个"页面权重库"来纠正陷，页面权重库是一个假想的节点，它与图中所有其它节点双向连接。陷一般存在于分布式超文本系统中；因此，涉及在有向图上随机漫游的每种方法都必须以某种方式处理这个问题。

基于IBM的CLEVER项目所做工作的另外一种方法已被Cornell大学(美国)的Jon Kleinberg申请专利(美国专利No. 6,112,202，其内容在此通过参考被合并)。该算法常常被称为HITS("超文本引导主题选择")。

在HITS方法中不存在已知的陷问题，因为在应用HITS运算符*BF*或者*FB*中的任何一个时，人们在顺着那些箭头(有向弧)以及逆着它们移动这两种动作之间交替。在多个专利(例如美国专利6,112,203，6,321,220，6,356,899和6,560,600，其内容在此通过参考被合并)中阐述

了这种方法及其变体。

一个有13个节点(文档)的简单图形如图2所示。在图2中，有两个陷区域：一个陷区域包括节点组(6,7,8)，而另外一个陷区域包括节点组(10,11,12,13)。仅仅顺着这些箭头的任何移动一旦先到达任一个陷区域，都将陷入该区域中。

陷的存在对于通过链接分析的重要性排序提出一个很大的实际问题。这个问题是：对于某些方法，陷节点或陷区域可以累积所有的权重，而其它非陷节点(文档)获得零权重。这样一来，在整个有向图上不可能获得一个稳定的非零正权重分布。没有这样一个权重分布，则一个有意义的文档排序变为不可能。也就是说，文档典型情况下经由链接分析被排序：通过计算每个节点的非零正“重要性权重”——从邻接矩阵的选定修改的主特征向量中获得——然后使用该'链接分析重要性权重'以及其它重要性度量(比如文本与查询的相关性)来计算一个总权重，这个总权重接着给出文档的排序。当这里存在陷时，这个主特征向量易于在这张图形的大部分区域上具有零权重。基于这样一个特征向量的重要性排序失去作用。

从数学上看，说一幅图形具有陷相当于说该图形不是强连接的。一个有向强连接图是这样的一个图形：对于图形中的任何一对节点u和v，都存在至少一条从u到v的有向路径以及至少一条从v到u的有向路径。这些路径未必是穿过同一组图形节点。更通俗来说：在顺着有向链接移动穿过一个强连接图时，人们可以从任何开始位置到达任何节点。陷节点或陷区域的存在妨碍了这种情况：人们会“受骗”到陷中，再也出不来。因此，具有陷的任何图形都不是强连接的，因此对陷问题的任何补救都想使整个链接图变为强连接的。

Google的PageRank算法通过增加从每个节点到任何其它节点的一个链接来补救陷问题。就是说，对于图中的每个节点，增加一个被给了一个小权重的出链接(outlink)到所有其它节点。这种修正被称为'随机冲浪'运算符，因为它模仿了可以从任何页面(节点)随机跳到任何其它页面的网页冲浪者的效果。

理论上，当随机冲浪运算符被使用时，原始链接图被一个完整图结构扰乱。一个完整图是一张从任何节点到图中任何其它节点具有一个有向连接的图形。链接图被完整图扰乱导致一张还是完整的新图形。陷问题因此被解决——因为新的图形是强连接的——并且因此对于所有的节点确保一个整体排序。可是，这并非毫无代价而来。要付出的代价就是：牺牲链接图的稀疏结构并被一个稠密的新扰乱的图形所代替。这会导致两种可能类型的问题：1) 当矩阵是稠密的时候通常用于计算排序的算法变得更费时，以及2)链接图的结构改变。

第一个问题对于PageRank方法并不出现。由于随机冲浪运算符的特殊(完整并且对称)结构，它的效果可以很容易地被计算。因此，PageRank算法的计算时间不会由于增加完整图结构的而显著增加。

第二个问题还在。当然，不以某种方式改变图形的结构就把一个非强连接图改变为强连接图是不可能的。可是，实际上PageRank修改很大。也就是说：假设原始图很大——假设它有一百万个节点。(在环球网中的文档数目以十亿为单位。)于是，说这张图是'稀疏的'就是在说图中的链接总数大概与节点数目（在这种情况下是几百万）成比例。(这个数字是平均节点程度) 。可是在执行PageRank修改之后，链接数目是一百万乘一百万——大约一万亿左右。

图3示出了随机冲浪运算符对图2的图形的影响。在这里，只示出了被加到节点1的出链接。也就是说，在增加随机冲浪链接之后，节点1有12个出链接而非2个。图形中的所有其它节点也将有12个出链接。图2中的所有其它冲浪链接没有画出，只是为了避免视觉混淆(对于这张图形总共有135个随机冲浪链接。)

简而言之，PageRank陷补救涉及把一个可能很大数量的新链接加到原始图上。这种改变虽然在某种意义上很大，但是至少可以通过向所有增加的链接给出相等的权重来以一种不偏斜的方式而被执行。目前公开的方法还想同样以一种不偏斜的方式但是通过只是增加少数链接到原始图上来使图形强连接。

被WiseNut搜索引擎使用的另外一种算法(美国专利申请2002-

0129014)有点类似于PageRank。WiseNut方法(被称为WiseRank)通过把每个节点双向连接到一个"页面权重库"(被表示为R)也增加了大量的链接。这使得每个节点达到所有其它节点；并且实际上，在该算法中，两个跳跃u→R→v被挤压为一个。因此，在拓扑上，这与PageRank相同。可是，使用通过R的跳跃的可能性在WiseNut规则中是不同的——具有较低输出度的节点具有使用R的更高可能性。虽然如此，从专利申请中它显出：以与PageRank矩阵中找到的相同的方式，不稀少的结果的WiseNut矩阵较易管理。因此，对于WiseNut，同样的优势与劣势存在，正如上面对于PageRank所述的那样。

基于IBM的CLEVER项目所做工作的链接分析的第三种方法是Cornell大学(美国)的Jon Kleinberg的(美国No. 6,112,202，其内容在此通过参考被合并)。该算法常常被称为HITS("超文本引导主题选择")。HITS算法没有直接使用邻接矩阵；相反，它使用复合矩阵，如此构造复合矩阵以避免有陷。因此，可以说HITS方法包括一种用于避免陷问题的方法。可是，复合矩阵有它们自己的问题。其一，它们可以给出一个'有效图形'，它在原始图中链接的节点之间没有连接。在有些情况下，这会通向一个已连接的原始图，导致一个断开的实际图形。这无法获得断开图的一个有意义的整体重要性函数；因此在这种情况下那么需要进一步假设或修改。

复合矩阵也将连接在原始图中未连接的许多节点对。因此，在复合矩阵中有比原始邻接矩阵中更多的非零条目。可是，实验研究暗示这些复合矩阵仍然比较稀疏：在一个示例中，对于原始邻接矩阵中的每个节点平均大约有8个链接，在每个节点的有效图形中找到有大约43个链接。因此，HITS方法看上去同样给出了一个易管理的数字计算。

最后，使用复合矩阵几乎还没有商业使用，而非复合的PageRank方法已经获得了巨大成功。在申请人自己的试验(未公开)中，HITS方法给出了相当差的结果，而PageRank与美国专利申请10/687,602的方法二者都给出了优良的结果。(在这些测试中，'优良的结果'是指对最佳节点给出一个高排序。)因此似乎HITS和相关方法虽然计算精致但

是在排序方面没有给出优良的性能。本发明人发现的所期望的一个特征是一种不依赖于使用复合矩阵的方法。

发明内容

鉴于超文本链接分析的目前可用方案的上述缺点，本发明的一个目的是提供一个基于规则的方法以及相应的系统和基于计算机的产品，用于排序超级链接网络中的文档。

如上所指出，通常的有向图结构不是强连接的。具体地说，它们会有陷节点和／或陷区域，这在执行链接分析中引起问题。可是，典型的有向图具有强连接的组件(SCC)。一个强连接的组件只不过是一组节点(通常不是整个图)，对于这些节点，总是有一条从任何节点u到任何其它节点v的路径，只要u和v位于同一SCC中。

在不同的SCC之间也有链接。可是这些链接总是单向的。这是由于如果在SCC_1和SCC_2之间在两个方向上有定向链接，那么两个SCC实际上只是一个。

本发明提供两个新的解决技术问题的方法，该问题在人们试图执行基于链接分析的排序时出现——即，陷问题。具体地说，本发明建议了用于解决陷问题的两个新方法。这两个方法每一个都具有两个所希望的设备：

- 它们适合与任何类型的(正向，后向，标准化的，非标准化的)非复合矩阵一起使用。
- 它们不把原始的稀疏图形改变为一个稠密的图形。相反，它们按照一种让图形稀疏的方式来修改图形。
- 方法1向原始图增加了少数的链接；而方法2没有增加新的链接。

附图说明

正如通过当结合附图考虑时参考如下详细说明变得更好理解那样，将很容易获得本发明的更完整评价以及它的许多附属优点，附图

中：

图1描述了一个一般的搜索环境；

图2描述了一个具有陷区域的示例图形；

图3描述了随机冲浪运算符对图2的图形的影响；

图4描述了与图2的示例图形对应的完整元图(metagraph)；

图5描述了与图2的示例图形对应的挤压图(collapsed graph)；

图6描述了方法1对图2的示例图形的影响；

图7描述了方法1对图4所示的元图的影响；

图8描述了方法2对图4所示的元图的影响；和

图9是与本发明相关联的一个计算机系统的方框图。

具体实施方式

本发明使用链接分析来计算每个节点u的节点链接分析权重LA(u)。为了排序目的，这是计算每个节点的一个文本相关性节点权重TR(u)的通常做法。一个最终节点权重W(u)然后可以作为这两个权重的加权和而被获得：

$$W(u) = a \cdot TR(u) + b \cdot LA(u)$$

因为权重W(u)被纯粹用于排序，并且只有比值a/b对排序有任何影响，所以两个参数a和b中只有一个是一个独立的调整参数。

包括在本发明中描述的那个方法在内的任何链接分析方法的出发点都是一个有向图，其中：节点是信息文档，并且链接是从一个节点到另一节点的指针。这张图形通常通过爬行或测量一组链接文档之间的链接而获得。我们将称这张图形为'已测量的图形'。

根据必须处理质量控制的各种准则来编辑已测量的图形常常是有用的。例如，如果确定为了人为升高一个或多个文档的排序的目的已经产生了大量的链接，那么这些链接可以被删除以便给出一个更精确且公平的排序。类似地，节点可以被删除；例如，如果多个节点有着几乎完全相同的内容，并且位于文档系统那同一区域中——，以使它们可以基本上被视为彼此的拷贝——那么这样的节点除了一个以外

全部可以被删除。当然，当节点被删除时，连接到这些节点的链接也必须被删除。

我们注意到这样的编辑总是采用修剪的形式：节点和／或链接的删除，以便增强导致的超级链接图的能力来精确表示链接文档组的实际结构。当用这种方式修剪已测量的图时，我们称结果图为'已修剪图'。

可以在链接分析中的任何阶段执行修剪。在开始链接分析之前检查并修剪图形始终是可能的。可是，链接分析进程本身揭露了关于节点和／或链接的质量信息也可能发生，这推动了进一步的修剪。因此，在链接分析期间的任何阶段修剪还是可能的，并且常常是所希望的。为此缘故，本发明允许在链接分析之前或链接分析期间修剪。

为了语言简洁并且在区别不是很重要的那些情况中，我们将经常使用名词'原始图'来指代已测量图或者已修剪图。在这里的要点是：本发明涉及原始图的修改使得消除陷问题。如果没有执行修剪，那么修改被应用到已测量图上；否则，修改被应用到已修剪图上。原始图然后正是通过本发明的方法修改了的那张图。已测量和已修剪图二者都可以通过一个相应的邻接矩阵表示。我们在这里使用协定：'邻接矩阵A'是指原始图的邻接矩阵。本发明的每个方法都修改这个矩阵。为了注释简洁，我们对于任一方法把如此修改了的矩阵表示为M_{SK}(在此，M_{SK}代表'陷补救')。

本发明使用'元图'的概念以便找到处理陷的新方法。对于任何给出的有向图，如下从原始图形成元图：

- 找到所有的SCC。
- 用单个'元节点'代替每个SCC。
- SCC内部的链接从而被忽略。
- SCC之间的链接保持不变。也就是说，从SCC_a中的某些节点到SCC_b中的其它节点的有向连接变成从元节点a到元节点b的链接。

结果元图没有循环——即(在此有向链接暗指有向流)；元图仅仅由一个方向上（从源到陷）的流组成。这样一个图被称为一个'有向非

周期图'(DAG)。

从图2的示例图形中获得的元图如图4所示。在这里很显然：任意流在一个方向上移动——从'源区'(1,2)经由中间区域到陷区域(6,7,8)和(10,11,12,13)。

在标准的文献中可了解被称为"挤压图"的一个密切相关的图形。除了它没有给出关于所有的SCC间链接的信息之外，它与元图相同。相反，它只是给出每对SCC之间的所有链接的方向(如果存在任何链接的话)。因此，挤压图是与元图相同的DAG，可是，每对SCC之间的所有并行链接被单个链接替代。元图和挤压图之间的这个区别在下面的讨论中将很有用。图2的挤压图如图5所示。

本发明合并了用于解决陷技术问题的两个新方法。这两个新方法都采用挤压图作为它们的出发点。因此，我们用三个步骤来呈现新的解决方案:

找到挤压图

方法1(反转链接)

方法2(泵压源)

找到挤压图

找到一个有向图的SCC是一种用可用的标准算法已经解决了的问题。这种解决方案的一个重要的方面是：它"随图的尺寸是线性的"——也被表示为"*N*阶的"或者更简洁地表示为O(*N*)。在这里，"图的尺寸"被视为节点(文档) 的数目，即*N*。这是指随着图形增大，解决这个问题需要的时间量只是随着图的尺寸(节点数，即：N)线性增长。SCC算法随图尺寸的这种缓慢增长对于目前公开的方法对大图的应用性很重要。许多文档系统具有巨大的文档数量——例如，环球网的尺寸现在估计大约为40亿个文档。因此，被应用到大图的任何方法都绝不需要随图尺寸*N*非常快速增长的计算或存储；并且线性的O(*N*)增长是当前可接受的科技发展水平。

涉及稀疏矩阵的任何方法计算节点权重将需要这样一个计算时间，该时间随图尺寸至少线性地增长。即：节点权重计算需要被邻接

矩阵重复相乘。如果邻接矩阵是稀疏的，那么它里面的非零条目数目将是O(N)——即，与图形中的节点数目成比例——并且因此每次乘法将要求与N线性增长一次。然后，如果迭代(乘法)数目根本没有随图尺寸增长，则总计算时间也将随N线性增长。有证据说明PageRank计算需要只是随图尺寸弱增长（即使有的话）的若干迭代(参见T. H. Haveliwala的Efficient Computation of PageRank，斯坦福大学技术报告，1999，其全部内容在此通过参考被合并)。因此，PageRank计算并且推断诸如在美国专利申请10/687,602中公开的方法之类的类似技术可能只是随图尺寸线性增长。

简而言之：节点权重计算所需要的时间随节点数目(图尺寸)N线性增长(或者也许比线性稍微快一点)。因此，找到已知只随N线性增长的SCC所需要的附加计算时间是完全可接受的。

SCC查找算法的存储要求也是可接受的。例如，Tarjan的算法需要存储所有节点，因此存储需求为O(N)。(参见Robert E. Tarjan的Depth-first Search and Linear Graph Algorithms，SIAM Journal on Computing，1(2)：146-160，1972，其全部内容在此通过参考被合并)。Tarjan的算法的一个改良版本需要甚至更少的存储。(参见Esko Nuutila 和 Eljas Soisalon-Soininen 的 On Finding the Strongly Connected Components in aDirected Graph，Information Processing Letters 49(1993) 9-14，其全部内容在此通过参考被合并)。

元图不但包含了有关所有SCC的信息；它还包括所有的SCC间链接。这个进一步的信息通常不可以从标准算法中获得。这些替代地给出“挤压图”，挤压图把所有SCC作为元节点(与元图相同)。可是，挤压图对定向链接的每对SCC通常只给出一个SCC间链接。(把示出完整元图的图4与示出挤压图的图5比较。)挤压图就这样示出了任意两个互相链接的SCC之间的流动方向；但是它没有对于本发明的方法给出足够的信息。因为两个公开的方法需要不同类型的附加信息(除挤压图以外)，所以每种方法被分别讨论。

方法1

方法1需要完整元图：它必须知道所有的SCC间链接(正如在图4的示例图形中看到的那样)，包括它们的开始点和结束点(可从图2中的完整图中获得)。

在典型情况下，稀疏有向图的邻接矩阵被存储为一个有序对列表。例如，如果链接$u \rightarrow v$存在于该图形中，则在列表中将有这样形式的一行：u v。

假设有两个SCC：SCC_1和SCC_2；并且假设人们从挤压图(从标准算法获得)中已知它们被链接。最后假设链接如下：SCC_1→SCC_2。然后人们知道这两个SCC之间的所有链接始于SCC_1并结束于SCC_2。人们(同样从标准算法中)还知道哪些节点位于SCC_1中并且哪些位于SCC_2中。因此，人们可以扫描有序对列表(即，稀疏邻接矩阵)，寻找始于SCC_1中的一个节点并以SCC_2中的一个节点结束的所有条目。在最坏情况中，这将花费的时间等于链接数目——对于一个稀疏矩阵，它与节点数目成正比，因此是$O(N)$级的。如果邻接矩阵被分类(这是很典型的情况)——以使始于一个给定节点u的所有条目被归组在一起——则人们通常将不需要搜索整个列表；但是所需的时间在任何情况下都为$O(N)$级的。

因此，存在一种简单的算法， 使用$O(N)$级的时间来查找所有的SCC间链接，因此将查找整个元图。给定完整元图，那么增加每个SCC间链接的反向不是问题。

例如：假设SCC_1和SCC_2有如下链接加入它们：

$u_1 \rightarrow v_x$

$u_2 \rightarrow v_y$

$u_3 \rightarrow v_z$

即，每个u节点位于SCC_1中，并且每个v节点位于SCC_2中。(注意：这两个SCC之间的所有链接具有相同的方向——与SCC_1和SCC_2是两个不同的SCC的假设一致。)

目前公开的方法于是建议通过增加如下链接使这两个SCC成为

单个SCC:

$u_1 \leftarrow v_x$

$u_2 \leftarrow v_y$

$u_3 \leftarrow v_z$

因此，在本发明的一个实施例中，每个SCC间链接(即，元图中的每个链接)补充一个反向链接。在这里，我们回想一下：元图是从可能已被修剪的原始图中形成而来。因此，元图中的SCC间链接可能都被视为'优良的'链接；并且因此一个不偏斜的方法要反转它们全部。

在一个替换实施例中，只有SCC间链接的一个子集补充了一个反向链接。即，对于每对已链接SCC只要反转至少一个 SCC间链接就可解决陷问题；并且有时候，人们可以反转甚至更少的 SCC间链接，而仍然使整个图形强连接。通过只反转SCC间链接的一个子集可能引起利用这个灵活性有利的时机。

此外，给这些陷补救链接一个权重ε是可能的，它可以被调整以便给出最佳性能。在一个实施例中，ε的值被保持低于原始链接的值(它通常为一)。用这种方式可避免太强烈地扰乱原始图。

再一次参见图2和6，它们示出了方法1对一个简单图的影响：每个SCC间链接被给出一个相反的伙伴；并且不需要其它新链接。

最后，一旦通过方法1已经使这张图强连接，则人们可以使用邻接矩阵的任何非复合形式(正向或反向，标准化或非标准化)来找到节点权重以及节点排序。

方法1对图2的示例图形的影响如图6所示。增加的链接是虚线。图 7示出了方法1对相应元图的影响，元图的未修改形式如图4所示。图 7更清楚地示出了方法1对于每个 SCC间链接只需要一个新的链接。即：对于这个简单图，方法1增加6个新的链接，而PageRank的随机冲浪运算符增加135个新的链接。对于极大的图形，两个方法之间的这种区别变得巨大。

方法2

方法2需要与方法1所使用的信息稍微不同的信息。方法2像方法1

那样始于挤压图。每个这样的图形是一个非循环有向图，或者说是DAG。这样的图形始终有至少一个源和至少一个陷。如果权重被置于源(元节点)处，并按照元图中的箭头方向移动，则这些权重将从源流向陷，一般除了陷元节点之外，在任何元节点处剩下零权重。

可是，在图形邻接矩阵的行动下，权重被一个给定因子放大。对于诸如PageRank之类的标准化方法，这个因子小于等于一；而对于诸如Canright和Engø-Monsen在美国专利申请10/687,602中公开的方法之类的非标准化方法，这个放大因子通常大于一。不管在哪种情况下，被视为与所有其它SCC(即，忽略全部 SCC间链接)隔离的每个SCC有一个给定放大因子或者'增益'。

如果元图中所有源SCC的'增益'(i)相等，并且(ii)大于任何其它SCC的增益，那么权重流可以达到一个平衡分布，并且在图中每个节点处有一个正权重。换言之，当这两个条件保持时，则邻接矩阵的主特征向量在各处都为正。这个论点的证明可以在申请人在2004年 5月2日递交给JACM将要发布的文章"Importance Functions for Directed Graphs"中可以找到，其全部内容在此通过参考被合并

总地来说，条件(i)和条件(ii)都没有保持。方法2通过调整位于源SCC内的所有链接的权重来强迫两个条件都保持。这样做时，人们调节源SCC的增益，直到两个条件(i)和(ii)都满足为止。

方法2然后包括如下步骤:

首先，人们取用元图中的每个SCC，并且忽略把这个SCC连接到任何其它SCC的所有链接。即：每个SCC被认为是与其它SCC相隔离的。该方法然后对于每个SCC需要所有的SCC内链接——位于SCC内部的所有链接。这些可以通过一个基本上与方法1中被用来查找 SCC间链接的过程相同的O(*N*)过程来查找。这然后为每个孤立的SCC给出完整的邻接矩阵。

人们然后使用期望形式的邻接矩阵(正向或反向，标准化或非标准化)来计算每个(孤立)SCC的增益(主特征值)。选择在这个步骤中使用哪个矩阵由这样一个矩阵来规定：该矩阵在计算重要性特征向量(参

见下面)时被用于整个图形。即：在每个步骤中必须使用相同的矩阵类型。

接下来，人们确定哪个SCC是源SCC。在比O(N)更少(通常少很多)的时间内很容易地从挤压图中获得该信息：源SCC是在挤压图中只有出链接的那些元节点。

如通常情况那样假定有一个以上的源SCC；这些源SCC有不等的增益；并且至少一个源SCC的增益小于某个其它SCC的增益。然后目的是增大所有源SCC的增益，直到满足如下两个条件：(i) 所有的源SCC必须具有相同的增益；和(ii)源SCC的公共增益因子必须大于任何其它SCC的增益因子。

具体地说，假设一个给定源SCC具有增益g，并且人们希望把它的增益增加到$G>g$。在该实施例中，这里有这么做的一个不偏斜的方式如下：把给定源SCC中的所有原始的内部链接权重(其同样通常为一)乘以因子G/g。这个简单的变化将造成期望的效果。在源SCC由单个节点组成并因此没有内部链接的特殊情况中，人们可以增加从该节点指向它自己的一个'自链接'并且给这个链接一个增益G。

通过为所有的源SCC选择相同的增益G，同时确保G大于任何其它(非源)SCC的增益，则条件(i)和(ii)被满足。然后，如三个发明者(下面引用)的预印本中所示，完整的修改图的主特征向量(正如在这里详述的那样，唯一的修改是所有源SCC中的内部链接权重的调整)在图形各处将为正。因此，这样一个特征向量可被用作节点的重要性度量。

用这种方法，通过使G只是比在所有非源SCC之中找到的最大增益g_max稍微大一点，对原始图的干扰可以被保持得尽可能小。

在这里应该再三重复：在完整的修改图上用来查找整个图的重要性度量的矩阵必须是与被用来查找每个孤立SCC的增益的矩阵具有相同的矩阵类型(正向或反向，标准化或非标准化)。

在本发明的一个替换实施例中，只对源SCC的一个子集调整(增大)增益。可是常规规则仍然保持：被泵压（pumped）的源SCC的公共增益必须被选择为大于任何非泵压的SCC（不论是源SCC与否）的

未修改增益。这么做的效果——如发明人的预印本中所示——是向非泵压源SCC中的全部节点给出零权重。同样，对于权重流只(直接地或间接地)依赖于非泵压源SCC的任何SCC也将得到零权重。尽管如此，对于某些源SCC，可能希望这么做。例如，假设一个源SCC由指向一个或多个SCC的单个节点组成，并且这些SCC可以从其它源SCC中得到权重。这个单个源节点因此指到图形的其余部分中(文档组)；但是没有文档指向这个节点。因此相对于完整的链接文档组，这个节点(文档)可能被判断为具有非常小的重要性。给这个节点一个增益G使得这个小源SCC注入和其它源SCC同样多的权重到图中。可能允许这样一个很小、几乎孤立的SCC注入那么多权重被判断为没有给出最佳结果。然后，在方法2的替换实施例中，人们可以选择不泵压一些源SCC——从而在得到的链接分析中给它们零权重。对于具有许多这样很小的、几乎孤立的源SCC的图形，使用本发明的方法1代替使用方法2可能有利。

如上所指出，选择不泵压一个源SCC导致向该源SCC中的节点给出零权重。这还意味着该SCC没有注入权重到图的其余部分中。因此，非泵压的源SCC可被认为已从图中被修剪掉。此外，在希望不泵压源SCC C_x——可是它的未修改增益g_x大于图形中任何其它SCC的增益——的情况(也许不太可能)中，人们面临两个选择而非一个。即，人们可以将所有其它源SCC泵压到增益$G>g_x$；或者，人们可以只从图中修剪掉该SCC C_x。后一个选择允许把剩余的源SCC泵压到一个较低的增益，因此对原始图修改较小。这些选择的任何一个都具有向非泵压的SCC C_x中的节点给出零权重的效果。

可能希望不向这样的一个或多个SCC给出零权重：这些SCC是一个非泵压的源SCC的'下游'，而且对于权重，它们只依赖于非泵压的源SCC。例如，如果人们判断图3中的源SCC(1,2)不重要，并且选择不泵压它，那么下游的SCC(3,4,5)(并且在这种情况下，实际上是所有的其它SCC)将获得零权重。我们可以说在这种情况下，不泵压源SCC(1,2)使得SCC(3,4,5)成为一个'有效源'。更精确地：一个SCC是一个有效源，

如果(i)对于权重，它只依赖于非泵压源SCC；并且(ii)如果从元图中修剪掉非泵压源SCC则变得一个源SCC。这个定义很有用，因为它建议了方法2的另外一个实施例。人们可以选择泵压任何有效的源，按照被用于任何其它源SCC的相同准则给它一个增益G。即：在图3中，如果人们选择不泵压SCC(1,2)，那么人们获得泵压SCC(3,4,5)的选择。

关于方法2应该澄清另外一个要点。即：诸如PageRank方法之类的某些方法使用一个规范化矩阵(如上所述用全矩阵修改了的)来计算节点权重。把一个图形的邻接矩阵标准化有这样的影响：具有陷的所有SCC增益为1，而所有其它SCC(包括源SCC)增益小于一。因此方法2只能以失去严格的标准化属性作为代价被应用到这样的情况——因为对于方法2，源SCC的增益必须被设置为大于1的某些值G。可是增益G只需要稍微大于一；因此人们可以认为严格的标准化的变化很小。在这种意义上讲，人们可以把方法2应用到标准化以及非标准化矩阵上。

图8说明了方法2对图2的示例图形的影响。注意：没有增加新链接。相反，单个源区(1,2)的'增益'被增强，用大阴影圆形表示该区域。对于一个具有多个源区的常规图，在应用方法2之后所有的都将具有相同的增强增益。

个性化

当前，对"个性化"搜索存在更大的兴趣。即：寻求这样一个搜索服务，该服务对于同一查询向每个用户不给出相同的答复，而是给出一个以某种有用的方法适合每个用户兴趣的答复。进一步简言之：对于个性化的搜索，查询答复应该取决于查询以及谁在查询。

目前，在找到个性化搜索的好方法的竞赛中还不存在明显的领先者。同样，还存在大量的各种方法。在此可以把注意力集中在一个自然并且容易遵循PageRank和WiseRank的陷补救方法的方法，以便指出本发明的陷补救还把自身容易引导到一个类似的个性化类型。

如上所述，PageRank和WiseRank用它变成稠密的这类方法来修改给定的邻接矩阵-事实上，它具有从所有节点到所有节点的链接。然

而，所添加的链接用这样一个方法被权重，即它们的影响可以根据简单地向原始的未修改邻接矩阵的乘法结果添加一个权重列表(向量)。

即：假定M是邻接矩阵的期望形式，而M'是由所添加链接形成的矩阵，因此被修改矩阵的最终形式是M+M'。然后，对于每个节点的权重寻找通过使用重复乘以被修改矩阵而被完成。即，权重x的试验向量被重复地乘以矩阵M+M'，直到该权重向量收敛成一个稳定模式为止。

对于PageRank和WiseRank，被添加链接的矩阵M'具有这样一个形式，即大多数或所有的乘法可以被脱机完成：

$$(M + M')x = Mx + s$$

即，权重s的补充向量可以不用稠密矩阵M'来进行矩阵乘法而被计算。

对于PageRank和WiseRank的非个性化型式，补充向量s向每个节点(文档)添加相同的权重-然后如上所述，其主要效果是防止权重在陷区域中被陷入。然而，一个简单的个性化方法出现：人们可以偏置补充向量s，并且用对于每个搜索器都定制的方法来这样做。例如，如果搜索器具有一个兴趣档案P，(例如)被表示为一个具有每个关键字的权重的关键字列表，则每个文档u可以被给定一个分数P(u)(用已知的文本相关方法)。该分数表示文档有多匹配用户档案，并且对于每个用户和每个文档只需要被计算一次。然后，这些分数可以被用来形成一个个性化的补充向量：人们可以简单地将分数P(u)用于补充向量的第u个条目。这类个性化的补充向量的应用将产生更多的权重被给予在最终(收敛)权重中分数P(u)较高的页面。

因此，在PageRank和WiseRank方法中都出现的补充向量s可以被个性化，从而给出了个性化搜索。

然而，本发明的用于补救陷问题的上述方法没有在它们自身中产生这样的补充向量。事实上，很少的(方法1)或没有(方法2)链接被添加到原始图。尽管如此，通过使用在邻接矩阵自己的链接上有可能个性化权重的事实，目前公开方法还允许应用几十种形式的P(u)。例如，

假定从文档*u*(具有分数P(*u*))到文档*v*(具有分数P(*v*))存在一个链接。在基于链接分析的排序中，每个链接都被看作是一种推荐。因此，对于个性化排序，根据"推荐者"(*u*)有多匹配用户兴趣来加权推荐（链接）是很自然的。因此，人们可以简单地通过分数P(u)来加权链接。

别的可能性将使用关于节点的分数可用的所有信息。即，人们可以不仅通过指向节点u的个人兴趣分数，而且还通过被指节点v的个人兴趣分数来加权链接。这样做的一个简单方法是通过和$(P(u)+P(v))$来加权每个u→v的链接。

其它变化也是可能的。通常在大多数情况下，人们可以选择$(P(u)+P(v))$的任何单调递增函数$f(P(u),p(v))$。如果递增x或y(或其二者)得出该函数f也增加的结果，则函数f(x, y)用x和y单调增加。因此，在其大多数一般形式中，本发明允许用$(P(u)+P(v))$的单调递增函数$f(P(u),p(v))$来加权链接从而个性化链接分析。我们注意到，这类个性化的计算负担只不过是需要计算分数自身。这个负担对于任何使用这类分数的方法来说都一样。

总结以上所述：每个节点(文档)u可以被给予一个个人兴趣分数P(u)，其标识该文档有多匹配一个单独的用户兴趣。然后，个人兴趣分数可以被用来偏置从u指向v的链接的权重。每个链接u-v都可以被给予权重P(u)；或者替换地，每个链接*u*→*v*都可以被给予权重$P(u)+P(v)$。其它规则也是可能的，反映了如果链接指向具有高个人兴趣分数的节点或从那里被指向，则它们得到高权重的一般规则。然后，结果的个性化邻接矩阵*A**(在链接上具有个性化权重)可以将标准邻接矩阵A(包括1和0的)代替为一个用于链接分析的起始点。即，个性化邻接矩阵*A**本身是个性化后向矩阵*B**；其转置是个性化正向矩阵*F**；并且这些的列标准化型式分别是个性化规范矩阵*b**和*f**。本发明的方法1或方法2可以被应用到任何个性化矩阵。

简而言之：用于节点的个人兴趣分数可用于重新权重图形的链接。然后，使用方法1或方法2的链接分析(经由主特征向量)给出可以被用来排序节点的节点权重LA(u)。用于节点的个人兴趣分数P(w)是

个性化的起始点，并且不应该与从链接分析中获得的最终节点权重相混淆。

其它的个性化形式可以与本发明相结合。在本发明的替换实施例中，被适当地权重的个人兴趣分数P(u)可以被组合到个性化的补充向量s中。即：我们可以用一个调节参数α来设置$s(u)=\alpha P(u)$。这个向量可以在乘法进程每次迭代时被添加到节点权重向量x:

$$x_{new}=M_{SR}x_{old}+s$$

在此，M_{SR}是用方法1或方法2规定的陷补救来修改的原始矩阵(正向或后向，标准化或非标准化)。然后，这个等式被一直迭代到节点权重x收敛为止；然后，结果给出了链接分析权重LA(u)。

这方法相当于向已修改的原始图添加一个完整图。因为这个事实，所以当使用这个形式的个性化时，人们事实上可以选择不使用方法1或方法2的补救；而是使用下面的一个迭代:

$$x_{new}=Mx_{old}+s$$

使用未修改的原始图M，加上前一段落中定义的补充向量。本发明的这个实施例不同于PageRank之处在于它使用非标准化运算符：M矩阵不是列标准化，并且补充向量s没有其条目之和的约束条件。因此，如Canright和Engø-Monsen在美国专利申请10/687,602中所公开的方法所述，这个方法可以被用来基于F或B运算符来个性化链接分析。

只要列标准化的约束条件被丢弃，其它形式的补充向量就是可能的。具体地说，人们可以令$s(u)=\alpha\sum_{v}P(v)x_{old}(v)$；并且另一个可能选择是$s(u)=\alpha P(u)\sum_{v}P(v)x_{old}(v)$。所有这些选择都将收敛成一个正权重组，因为它们都用链接上的正权重来表示一个完整图。具体地说——假定M=F，因此将发生权重的正向传播——选择$s(u)=\alpha P(u)$代表在所有指向u的链接上都具有权重$\alpha P(u)$的完整图；选择$s(u)=\alpha\sum_{v}P(v)x_{old}(v)$代表在所有从u指向的链接上都具有权重$\alpha P(u)$的完整图；并且选择$s(u)=\alpha P(u)\sum_{v}P(v)x_{old}(v)$代表在u和v之间的所有链接上都具有权重$\alpha P(u)P(v)$的完整图。我们注意到，这三个选择中的任何一个选择都可以和原始矩阵M或被修改矩阵M_{SR}的任何非标准化形式一起使用。(不同于第一选择的)后两个选择

不能与诸如PageRank之类的标准化(权重保持)方法一起使用。这个的原因是s的所有条目之和为一个(标准化方法所需的)常数的要求不可能适用于涉及补充向量s中的x_{old}的条目的加权和的那两个选择。

在本发明的另一个替换实施例中，个人兴趣分数p(u)在链接分析程序中根本没有被使用。相反，它们仅仅被添加到用于文本相关性的节点权重TR(u)以及来自链接分析的LA(u)以便为每个节点给出最终节点权重W(u):

$$W(u) = a \cdot TR(u) + b \cdot LA(u) + c \cdot P(u)$$

在此，系数a、b、和c是调节参数；但是因为权重W(u)被用于排序，所以只有这三个中的两个是独立调节参数。

图9说明了本发明的实施例可以在其上面实现的计算机系统1201。计算机设计在STALLINGS,W.的Computer Organization and Architecture（第4版，Upper Saddle River，NJ，Prentice Hall，1996）中被详细论述，其全部内容在此通过参考被合并。计算机系统1201包括用于传递信息的总线1202或其它通信机构，和与总线1202耦合以用于处理所述信息的处理器1203。计算机系统1201还包括主存储器1204，比如随机存取存储器(RAM)或其它动态存储器(例如，动态随机存储器(DRAM)、静态随机存储器(SRAM)和同步DRAM(SDRAM))，它们被耦合到总线1202以便存储将被处理器1203执行的信息和指令。另外，主存储器1204可以在处理器1203执行指令期间被用于存储临时变量或其它中间信息。计算机系统1201还包括只读存储器(ROM)1205或其它静态存储器装置(例如，可编程序只读存储器(PROM)、可擦编程只读存储器(EPROM)和电可擦可编程只读存储器(EEPROM))，它们被耦合到总线1202以用于存储处理器1203的静态信息和指令。

计算机系统1201还包括一个耦合到总线1202来控制一个或多个用于存储信息和指令的存储装置的磁盘控制器1206，比如磁硬盘1207和可移动介质驱动器1208(例如，软盘驱动器、只读光盘驱动器、读/写光盘驱动器、光盘库、磁带驱动器和可移动磁光驱动器)。存储装置可以用一个适当的装置接口被添加到计算机系统1201(例如，小型计算

机系统接口(SCSI)、集成电路设备(IDE)、IDE(E-IDE)、直接存储器存取(DMA)或超DMA)。

计算机系统1201还可以包括专用逻辑装置(例如，专用集成电路(ASIC))或可配置的逻辑装置(例如，简单可编程逻辑器件(SPLD)、复杂可编程逻辑器件(CPLD)和字段可编程门阵列(FPGA))。

计算机系统1201还可以包括一个被耦合到总线1202来控制显示器1210的显示器控制器1209，比如阴极射线管(CRT)，所述显示器用于向计算机用户显示信息。计算机系统包括诸如键盘1211和点取装置1212之类的输入装置以便与计算机用户交互作用并且向处理器提供信息。例如，指向装置1212可以是用于向处理器1203传递方向信息和命令选择并且用于在显示器1210上控制光标移动的鼠标、轨迹球、或指向杆。另外，一个打印机可以提供由计算机系统1201存储和/或生成的数据的打印列表。

响应于处理器1203执行诸如主存储器1204之类的存储器中包括的一个或多个指令的一个或多个序列，计算机系统1201执行本发明的一部分或所有的处理步骤。这类指令可以从诸如硬盘1207或可移动介质驱动器1208之类的别的计算机可读介质中被读入主存储器1204。多进程方案中的一个或多个处理器还可以被采用来执行主存储器1204中包括的指令序列。在替换实施例中，硬接线电路可以代替软件指令或与之结合地被使用。从而，实施例不局限于硬件电路和软件的任何特定组合。

如上所述，计算机系统1201包括至少一个计算机可读介质或存储器，用于保存根据本发明教学来编程的指令并且用于包含此处所述的结构、表格、记录或其它数据。计算机可读介质的例子是光盘、硬盘、软盘、磁带、磁光盘、PROM(EPROM、EEPROM、flash EPROM)、DRAM、SRAM、SDRAM或任何其它磁介质、光盘(例如，CD-ROM)、或任何其它光介质、穿孔卡片、纸质磁带、或其它具有孔模式的物理介质、(如下所述的)载波、或计算机可以从中读取数据的任何其它介质。

本发明包括被存储在任何一种计算机可读介质或其结合上的软件：用于控制计算机系统1201，用于驱动一个或多个用于实现本发明的装置，并且用于让计算机系统1201能够与人类用户(例如，打印生产人员)交互作用。这类软件可以非限制性地包括装置驱动、操作系统、开发工具和应用软件。这类计算机可读介质还包括本发明的计算机程序产品，用于执行在实现本发明的过程中执行的所有或一部分处理(如果处理是分布式的)。

本发明的计算机代码装置可以是任何可解释或可执行码的机构，非限制性地包括脚本、可解释程序、动态连接库(DLLs)、Java类和完整的可执行程序。而且，本发明的一部分处理可以被分布以用于更好的性能、可靠性和/或成本。

在此使用的术语"计算机可读介质"指的是参与向处理器1203提供运行指令的任何介质。计算机可读介质可以采用许多种形式，非限制性地包括非易失性介质、易失性介质、和传输介质。例如，非易失性介质包括光、磁盘和磁光盘，比如硬盘1207或可移动介质驱动器1208。易失性介质包括诸如主存储器1204之类的动态存储器。传输介质包括同轴电缆、铜线和光纤，包括构成总线1202的金属丝。传输介质还可以采用声学或光波的形式，比如在无线电波和红外线数据通信期间所生成的。

不同形式的计算机可读介质可以被包含来实现处理器1203所运行的一个或多个指令的一个或多个序列。例如，指令可以最初在远程计算机的磁盘上被携带。该远程计算机可以把用于实现本发明的所有一部分的指令加载到动态存储器中并且在电话线路上用调制解调器来发送这些指令。位于计算机系统1201局部的调制解调器可以在电话线路上接收数据并且使用红外发射机把数据转换成红外信号。一个被耦合到总线1202的红外探测器可以接收红外信号中携带的数据并且将这些数据置于总线1202上。总线1202把数据传送到主存储器1204，处理器1203从主存储器1204取回并且执行该指令。在处理器1203运行指令之前或之后，主存储器1204所接收的指令可以被选择性地存储在存储

装置1207或1208上。

计算机系统1201还包括一个被耦合到总线1202的通信接口1213。通信接口1213提供一个被耦合到网络链接1214的双向数据通信，网络链接1214例如被连接到局域网(LAN)1215或诸如因特网之类的别的通信网1216。例如，通信接口1213可以是一个连结到任何分组交换LAN的网络接口卡。作为另一个例子，通信接口1213可以是不对称数字用户线路(ADSL)卡、综合业务数字网(ISDN卡或调制解调器，以便向对应的通信线路类型提供一个数据通信连接。无线链接也可以被实现。在任何这类实施中，通信接口1213发送并接收携带表示不同类型信息的数字数据流的电、电磁或光信号。

在典型情况下，网络链接1214经由一个或多个网络向其它数据装置提供数据通信。例如，网络链接1214可以经由局部网1215(例如，LAN)或者经由服务提供商操作的设备向别的计算机提供连接，服务提供商经由通信网1216提供通信服务。例如，局部网1214和通信网1216使用携带数字数据流的电、电磁或光信号和相关联的物理层(例如，CAT5电缆、同轴电缆、光纤等等)。经由不同网络的信号和在网络链接1214上并且经由通信接口1213的信号(携带往返于计算机系统1201的数字数据)可以用基带信号或基于载波的信号来实现。基带信号传送作为描述数字数据比特流的未调制电脉冲的数字数据，其中，术语"比特"将被广泛地解释为意指码元，其中，每个码元传送至少一个或多个信息比特。数字数据还可以被用来调制一个诸如具有振幅、相位和/或频移键控信号之类的载波，所述信号在传导介质上被传播，或者经由传播介质作为电磁波被发射。从而，数字数据可以作为未调制的基带数据经由"有线"通信信道被发送和/或通过调制载波在一个不同于基带的预定频带内被发送。计算机系统1201可以经由(一个或多个)网络1215和1216、网络链接1214并且通信接口1213来发射和接收包括程序代码的数据。而且，网络链接1214可以经由LAN1215向诸如个人数字助理(PDA)、膝上型计算机或蜂窝电话之类的移动装置1217提供一个连接。

目的是在一个集中网络搜索引擎中排序比特的本发明实施需要

它与几个其它元件的结合：文本排序系统、索引系统、爬行检测器和用户接口。在这个实施中，本发明表示一个完整的工作搜索引擎一部分，和不能与这类系统的其它元件隔离而被实现。

本发明还可以被实现为在单个PC上保留的内容上操作的搜索引擎的一部分。这个实施需要在PC(即，"专用网络")上存储的所有文档(邮件、文本、演示文稿等等)之间引入超链接。在当今的操作系统中，这个主意(单个PC上的文档之间的超链接)只被实现到一个非常有限的程度。因此，把本发明实现为"专用网络"的一部分将需要更改PC中的许多文件处理应用。另外，索引系统、用户接口和(可能的)基于文本相关的排序系统将被需要。

根据上述教导，本发明的很多更改和变化都是可能的。因此应当理解，在要求保护的范围内，本发明可以除在此具体描述之外而被实践。

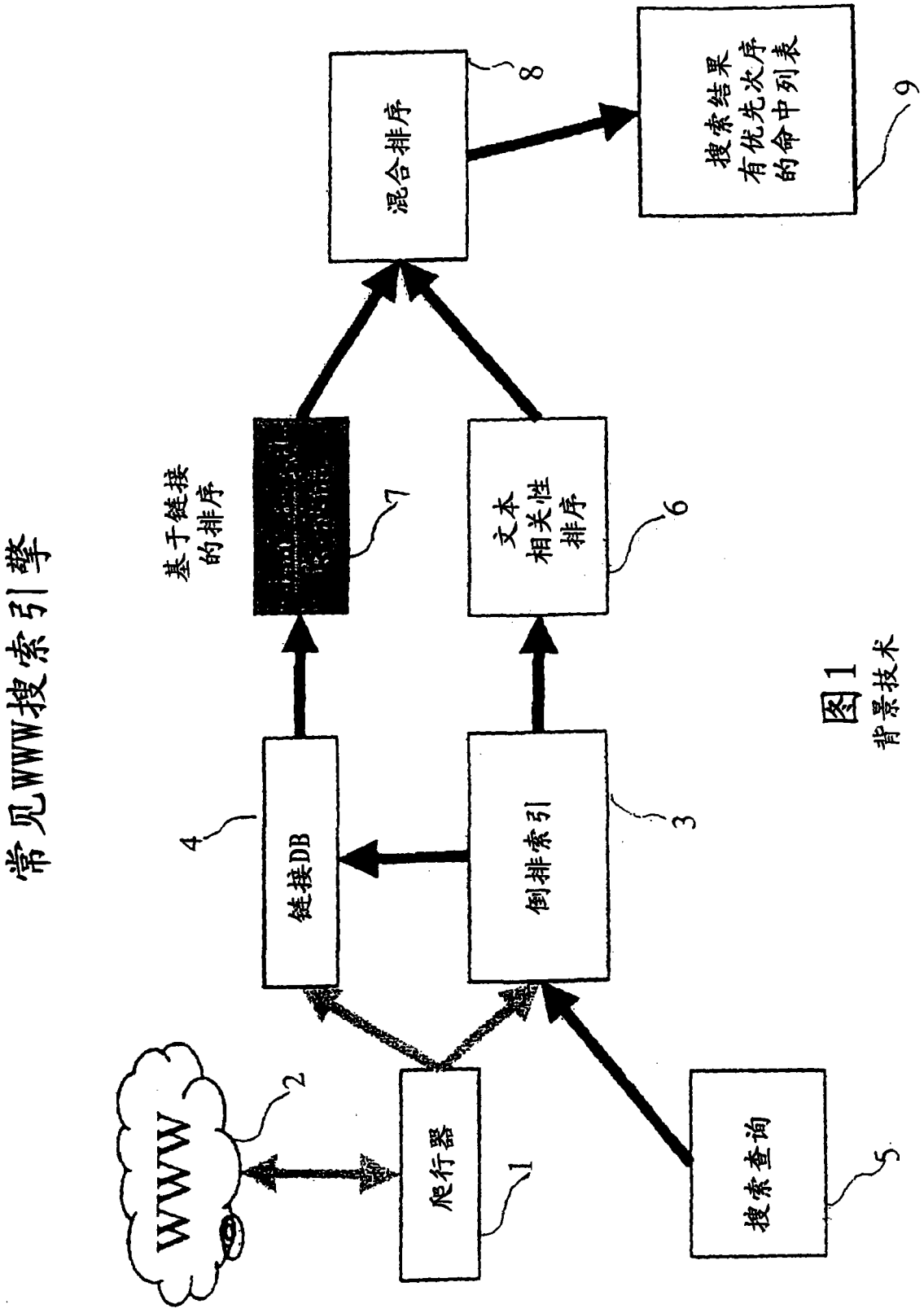

常见WWW搜索引擎
WWW
2
爬行器
1
链接DB
4
倒排索引
3
搜索查询
5
基于链接的排序
7
文本相关性排序
6
混合排序
8
搜索结果有优先次序的命中列表
9

图1
背景技术

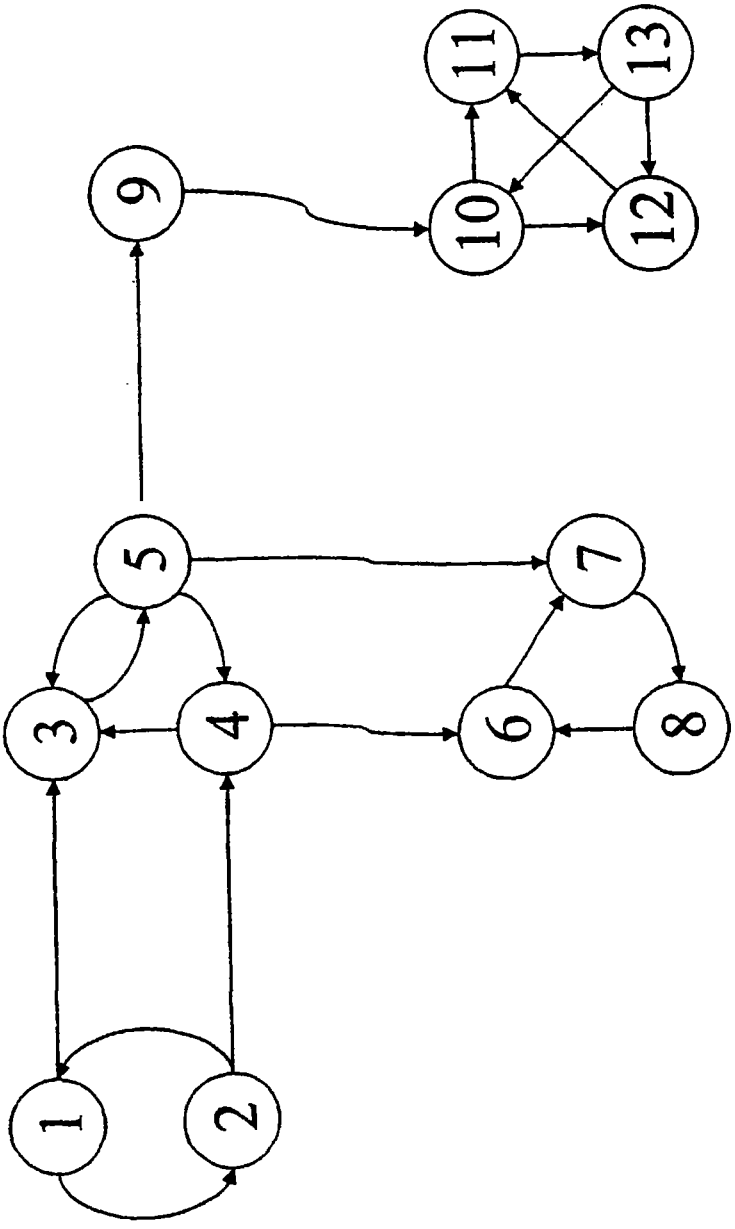


图2-有向图

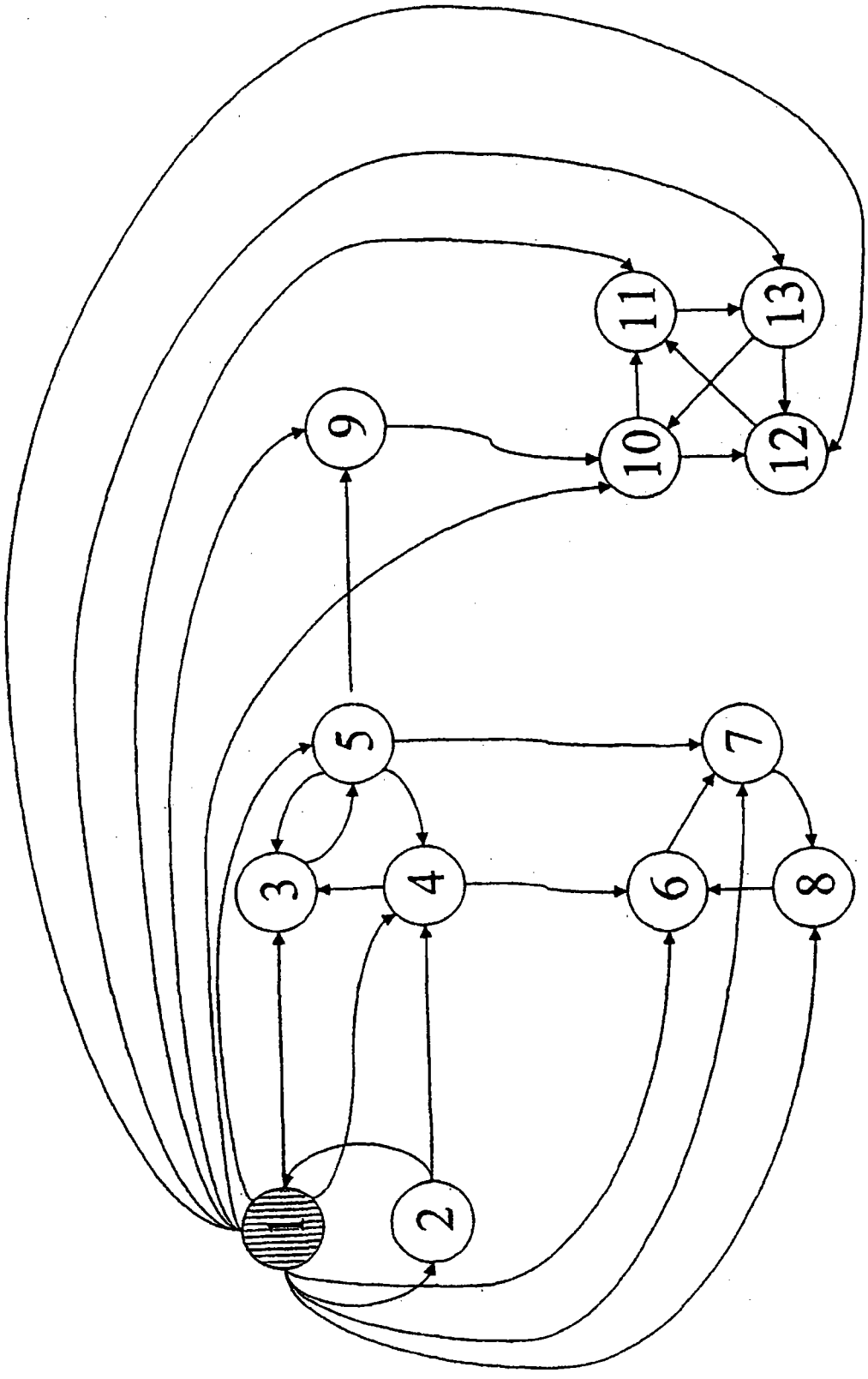


图3-随机冲浪

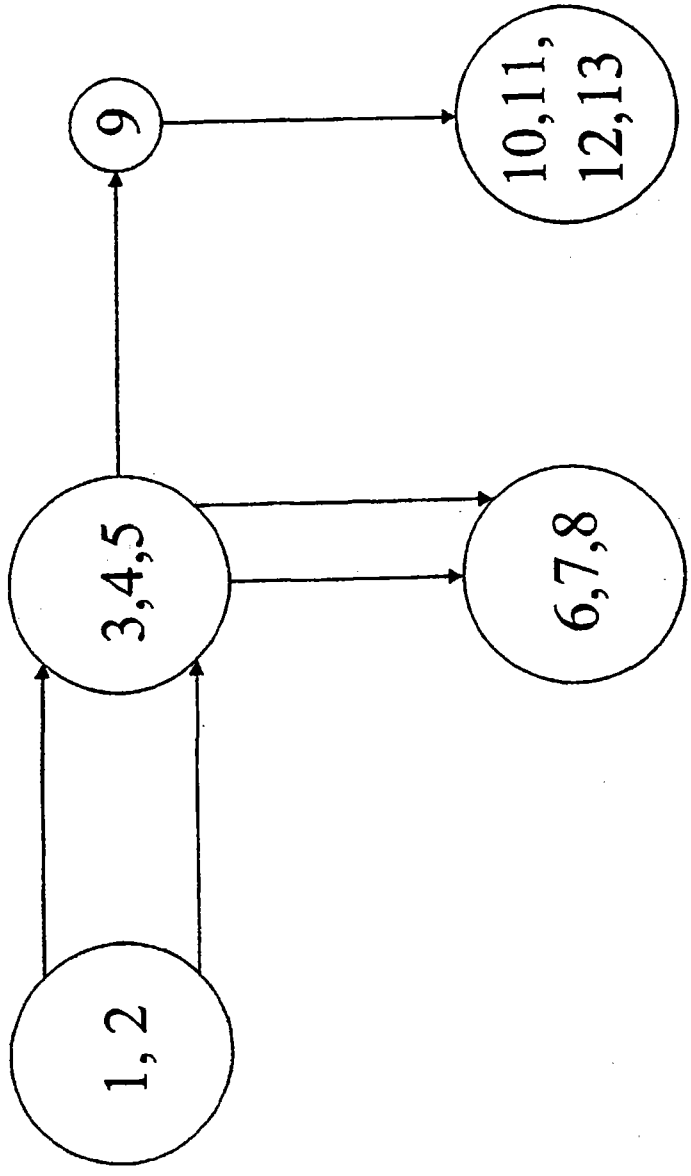


图4 -元图 (MG)

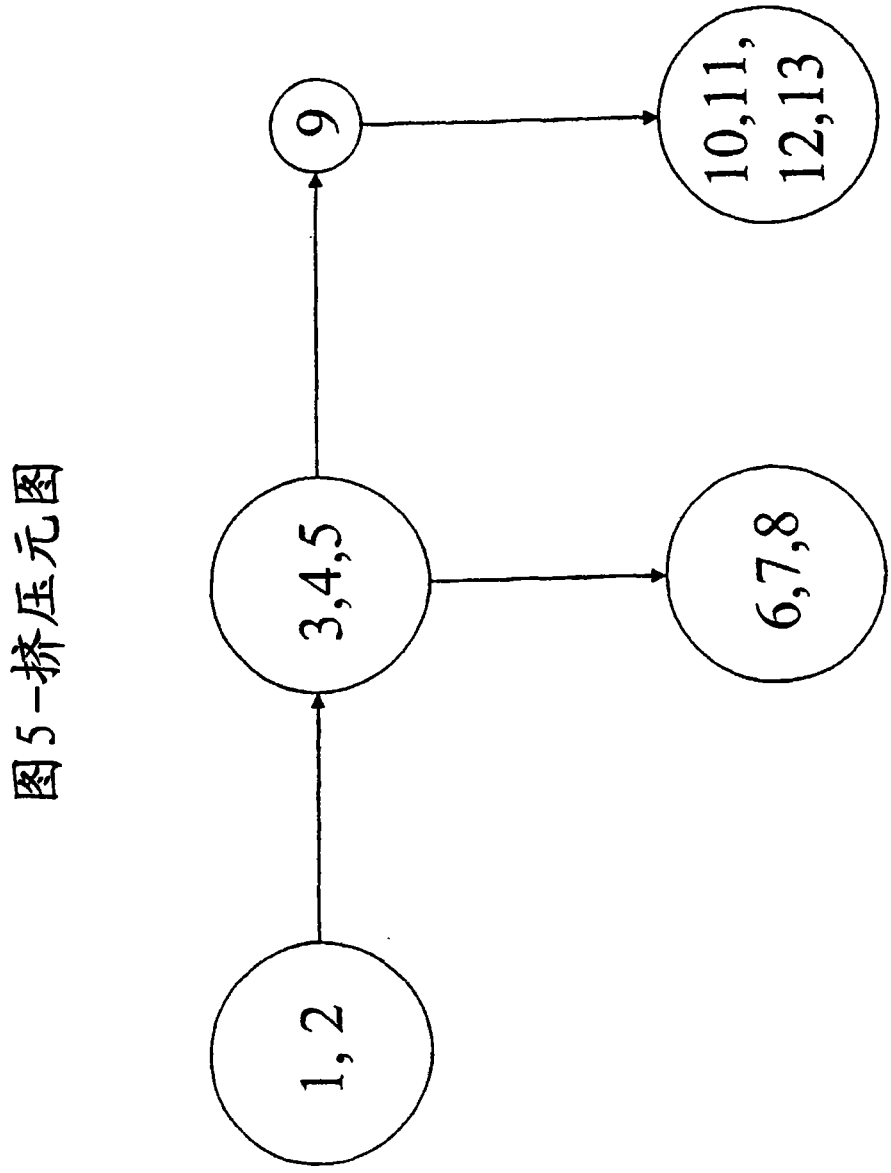


图5-挤压元图

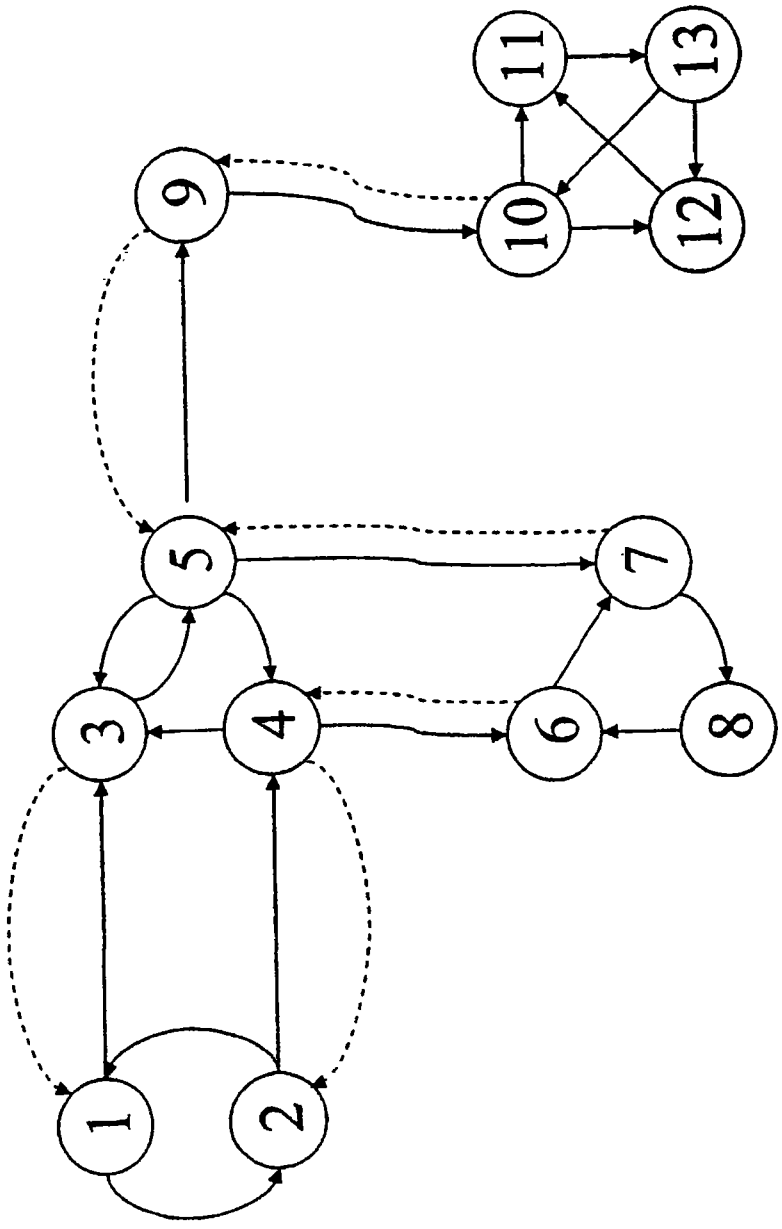


图6-图形‘反向’

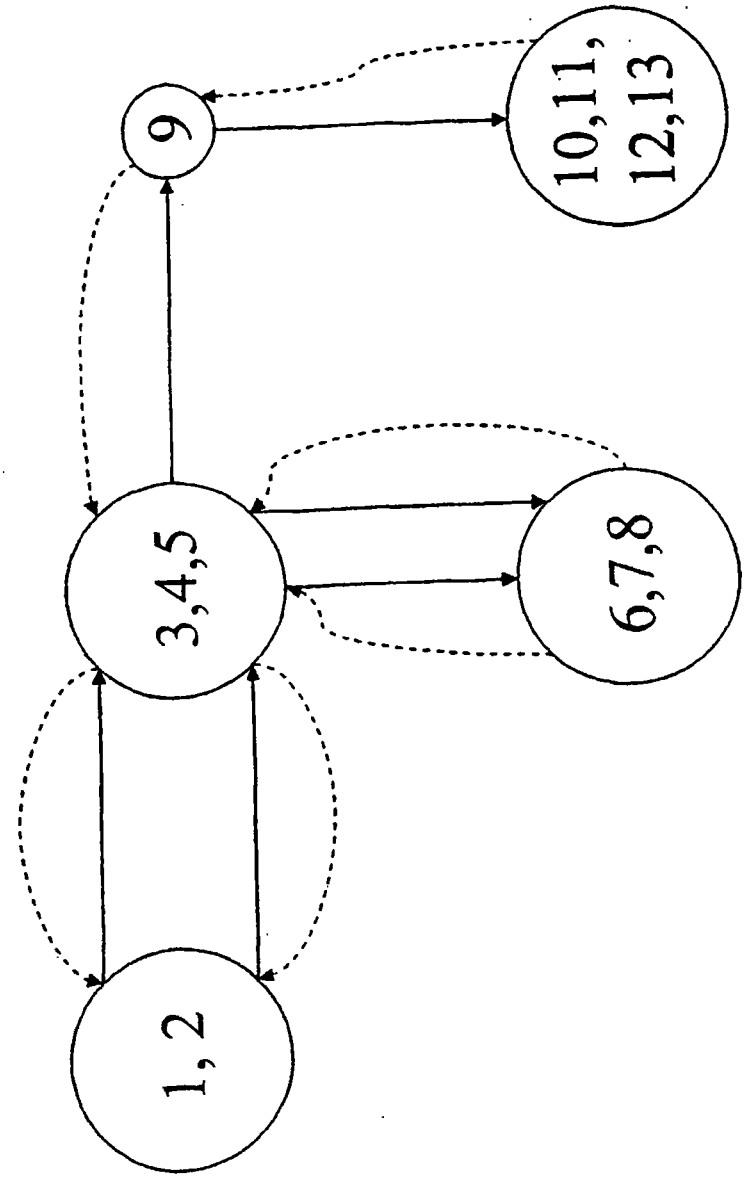


图7-MG‘反向’

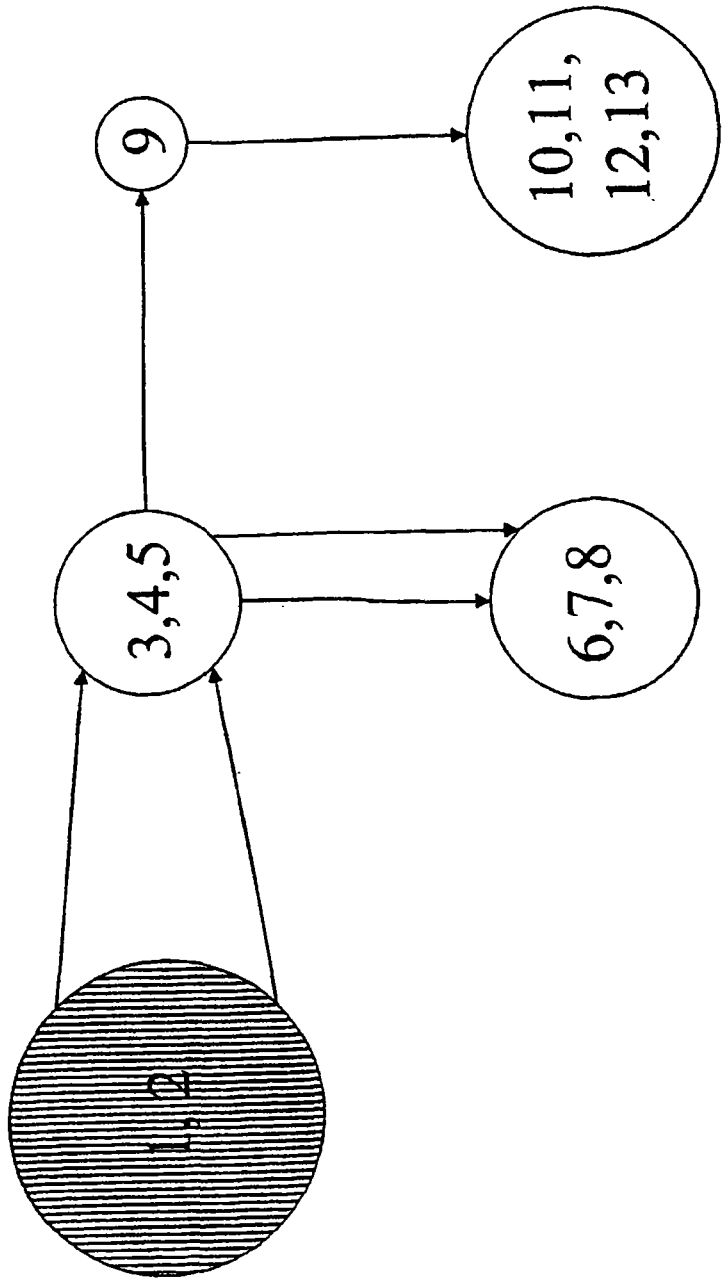
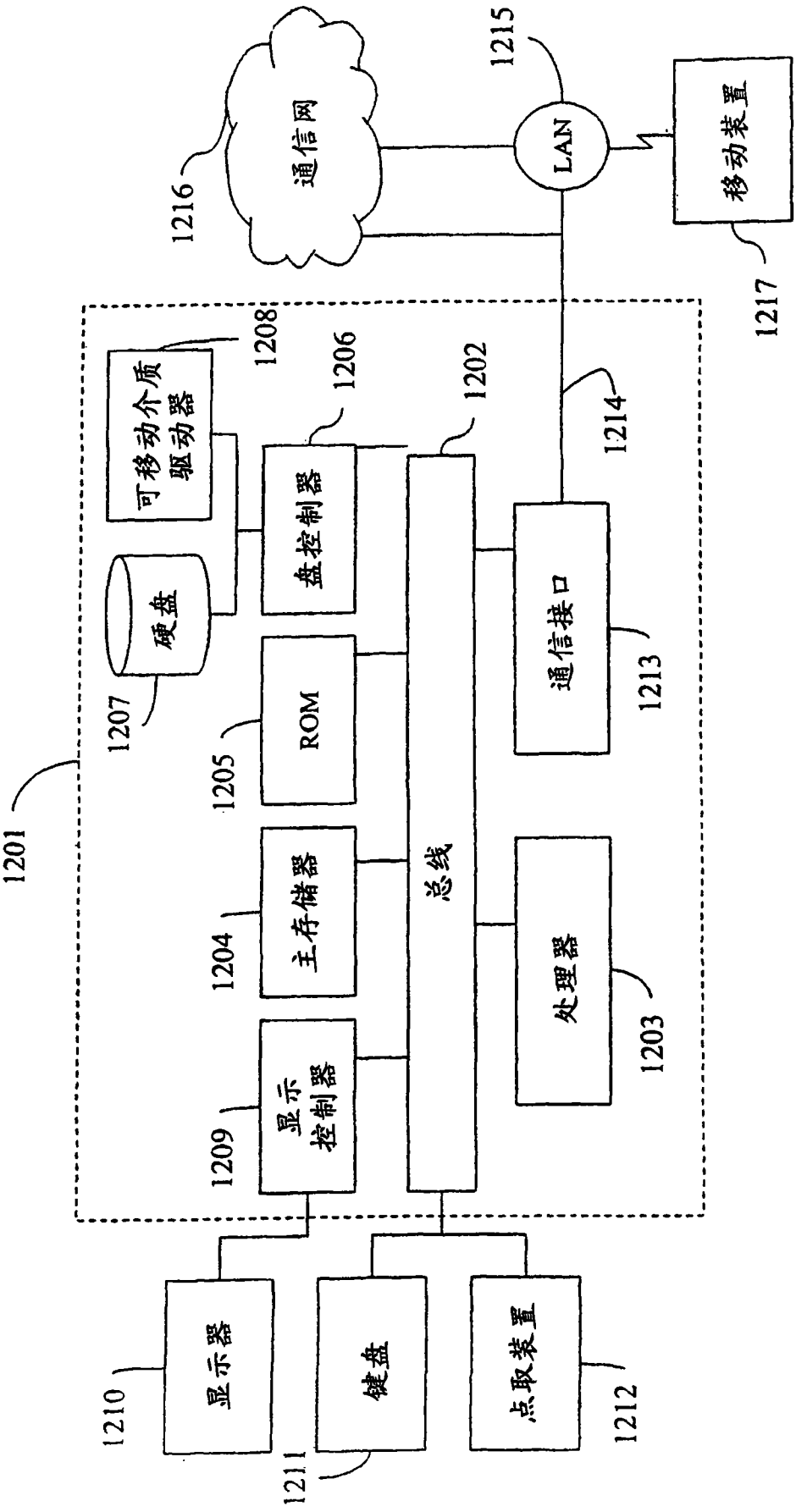


图8-MG‘泵压’

图 9


1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
显示器
键盘
点取装置
显示控制器
主存储器
ROM
硬盘
可移动介质驱动器
盘控制器
总线
处理器
通信接口
LAN
通信网
移动装置