DOI:10.13656/j.cnki.gxkx.1995.03.008
广西科学 Guangxi Sciences 1995，2（3）：28～33

# 利用簇化技术优化超文本数据结构
# Using Aggregation Clustering to Optimize the Hypertext Database Structure

黄 瑜
Huang Yu

（广西计算中心广西软件新技术实验室·南宁市星湖路32号 530022）
(Guangxi New Software Technology Lab., Guangxi Computing Center, 32 Xinghu Road, Nanning, Guangxi, 530022)

**摘要** 介绍一种新的方法，该方法能够从原始图结构中通过一种具有例外性的簇化（Aggregation Clustering with Exceptions）手段，获得高层次的关系。这种方法使用的是一个基于已扩展的 kernigher-line 算法的直接探索法。

**关键词** 簇化 例外 超文本数据库

**Abstract** With the hypertext and hypermedia use widely, the information to be managed in hypermedia system become more and more, and the hypermedia database become large and large. Extracting high-level structures is useful for providing a high performance browsing environment as well as efficient physical database design, especially when handling large amounts of data. This paper introduce a new method, ACE (Aggregation Clustering with Exceptions), which generates aggregations and exceptions from the original graph structure in order to capture high-level relationships. This method is based on an extended Kernighan-Lin algorithm.

**Key words** aggregation and clustering, exception, hypertext database

## 1 问题的提出

超文本已广泛发展成为对未来信息处理的一种媒体，它的导航界面——浏览和它的非常简单的结构——结点和链，允许用户更容易处理信息，在一些非均质和合作环境中，为结合复合超文本结构所使用的一种自底向上的组织策略是相当自然的。超文本结构的最大优点就是为用户提供导航路径。然而，在特定的系统中，这种优点仅被证实在相对小或非常稀疏的结构中，这里，将阐述当为大量可能密集的数据提供这样的导航时出现的问题的要求。

当处理大量的超文本数据，例如一个电子百科全书、技术手册和合作环境文档时，将会出现以下问题。

·迷路问题

用户不知道自己在信息空间的什么地方以及他们很难找到他们应该去的下一个地方[2]。

·缺少为相关联的链提供设计原理

两个结点之间是否要建立一条链，没有什么原则可遵循。可想而知，在一个有几万个结点中，利用手工将链建立起来是有多么困难！

·在构造花费和应用利益间的折衷问题

当采用一个简单的稀疏链策略时，用户被限制于只能在信息空间的很少部位做浏览，另一方面，当系统设计者试图提供一个很宽广的信息域中进行检索的技术时，组织和修改这样的系统的花费迅速提高，这种折衷问题存在于任何大小的系统中，特别是大系统[3]。

以上几个问题是超文本系统中所固有的问题，特别是对庞大的超文本数据库进行导航、维护时。因此，从庞大的超文本数据库中提出一个高水平的结构是解决问题的关键。本文所介绍的簇化技术能很大程度上提出一个高水平的结构，而且能进行高效的物理存贮。

1995-07-23 收稿。

## 2 带有例外关系性质的超文本关系的簇化技术

所谓的簇化技术，就是指将超文本数据库中，具有某种共同特征的链的结点归结为一个结点集合，对于哪些不能归结于结点集合的结点和它们的链则称之为例外集[1]。这里所介绍的技术就是具有例外关系的簇化技术。称之为簇化主要是将这些一个个结点集合称之为束，而这些结点集合之间常常还有链的存在，所以又称之为簇。

下面，通过两个例子说明这种的技术。并在最后一小节给出其 E—R 关系示意图。

### 2.1 例 1：学生—选课关系

让我们来看一个简单的例子：学生以及其所选的课程之间的关系。这个关系如图 1 (a) 所示，是一个多对多链接关系。学生选的课程越多，连接就变得越密集。然而，由于在实际系统中，有很多实际的链相关联。我们则通常通过适当的归并将链简单化。图 1 (b) 给出了对应于每个结点归并可能。同一个系的学生选的课程大部分是相同的，在某些情况下，很多基础课也都是学生要选的。例如，计算机科学系一般都要选数学系的课图 1 (b)。

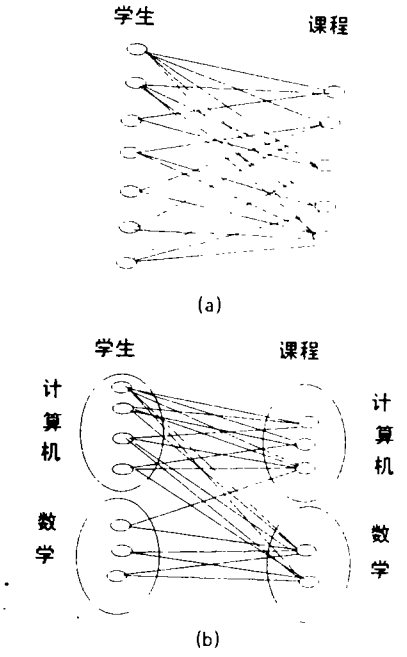


图 1 学生—课程的关系

Fig. 1 A student—course relationship

当然，同一系的学生所选的课也不见得都一定相同，当这些例外的学生是相对的少时，我们可以通过对物理数据库和对浏览示意图提供一个简单的表达方法。(以下所提到的“例外”都是指这种情况，即这样的链比较少，与其他任何归并链集中的意义都不相同，所以把它们单独处理，这些链称之为例外。)

### 2.2 例 2：一个文章引用关系

考虑在技术论文中的引用关系，如图 2 (a) 所说明的，一篇论文可以引用其他的论文。基础的论文和组织好的论文常常被引用。

通过适当的分解，由于相应的论文引用相类似的论文，结构可以简单化。如图 2b 显示的一样。可以强调全局的调用流。注意到在大部分的文章中同一范畴的，没有互相引用。然而，仍然存在调用和被调用关系。从时间顺序来说，后发表的文章引用前面发表的文章。

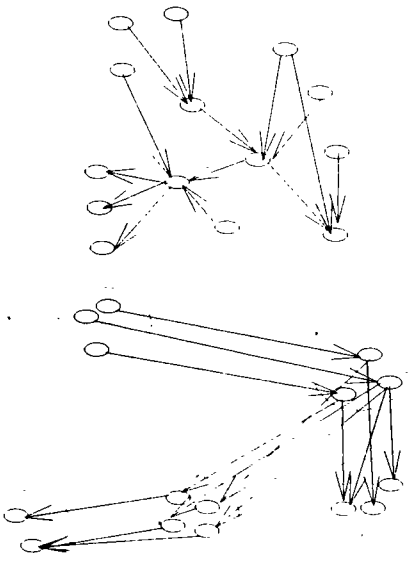

图 2 文章的相互引用关系

Fig. 2 A citation hierarchy

### 2.3 具有例外性质的簇化技术

这种技术的想法就是通过允许例外情况考虑一种有效的簇化技术，如图 3 所示的一样识别一个全程的关系。同诸如成块对角方法的相类似簇化相比。所提出的方法的主要特性k k 簇化技术，就是强调从实际的超文本形式中簇化联接的归纳，存在簇化方法的目的就是解除链接的链数量，也就是连接在簇化以后超文本数据库中的总和。因此，链的连接可能是不是一下子就能理解的。

考虑具有例外性质的簇化技术提供的一个更自然和提供资料的高级结构。这种技术要求必须平衡具有保持例外数量的簇化范围。这一图形压缩导致物理结构的更有效表达。另外，当有一个高效的浏览示意图的基本结构作为使用时，簇化以后的数据结构更容易使人理解，人类界面表达更加令人满意。

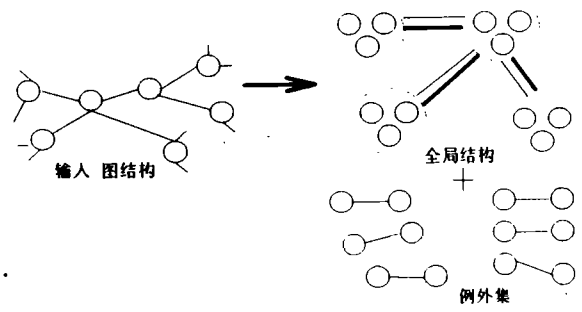


**图 3 具有例外性质的簇化**

**Fig. 3 Aggregation clustering with exceptions**

图 4 给出了一个 ACE 的 E−R 示意图。这是有两种类型的 ACE。它们取决于相关联的元素是否被拆散。例 1 是一个拆散的情况，即多对多关系簇化技术。例 2 是一个自相关联情况，即反身关系簇化技术。

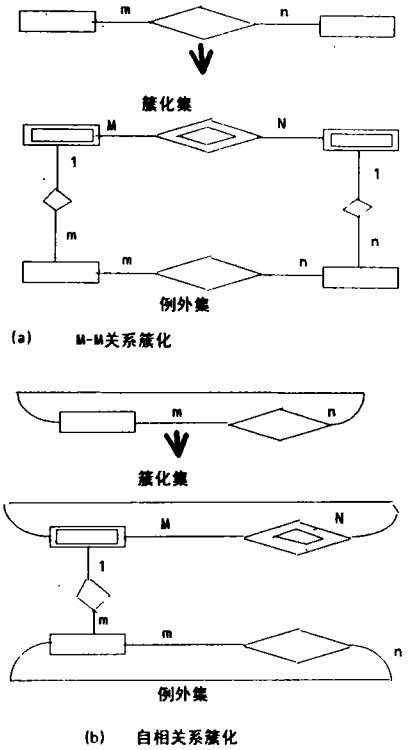


**图 4 具有例外性簇化的 E-R 示意图**

**Fig. 4 E-R diagram of ACE**

## 3 簇化技术的符号表示

在这一节中，介绍 ACE 的问题的定义并用例子来说明，这里只考虑相关联的超文本结构并用有向图来表示它们。其特定的算法在第 5 节讲。

假定输入到簇化处理系统中是一个有向图 $G=(V, E)$，这里顶点 V 表示结点，边 E 是超文本中的链。其输出的目标图有簇化技术图 $G_A=(V_A, E_A)$ 和例外图 $G_X$，它们合起来可以推算出输入图。

簇化后的图 $G_A$ 是一个超级图，这个图具备有超级顶点和超级边，集合 $V_A=\{V_{A1}, \cdots, V_{AS}\}$ 是 V 的划分，集合 $E_A$ 包含连接划分的边，$E_A$ $V_A \times V_A$。

例外图 $G_X$ 表示简化的簇化技术图 $G_A$ 和实际输入图之间的差异。有两种类型的例外。一种称为包含链，是一个没有相应超级边的次要输入边。图 $G_I=(V, E_I)$ 被称为是一种包含性的图。另外一种是专用链，在两点间没有输入边的时候，却有一条超级边。图 $G_E=(V, E_E)$ 被称为一个专用图。至于一条超级边是否生成主要取决于两个相应超级顶点大多数关系。如果它们之间有相当数量的临界级的输入边（临界级 $=0.5$），则会生成超级边。

提取这种簇化后的图和例外图使这些图变得尽可能的简易。在这种情况下简单的标准大概就是指图的大小。比如顶点的数量和边的数量，或单算边的数量。将原图转换成这样的简图，对于深透了解原始数据与压缩数据是有用的。

抽取一个全局的图解结构的问题定义描述如下

输入：输入图 $G=(V, E)$

输出：簇化后的图 $G_A=(V_A, E_A)$

包含图 $G_I=(V, E_I)$ 和专有图 $G_E=(V, E_E)$

下面，用 $V_{Ai}$ 表示簇化技术结点以及输入结点集合，从上下文来看，其意思可以理解，输入与输出满足以下限制：

$$V=U_i V_{Ai};\ \forall\ i, j, i\neq j;\ V_{Ai}\cap V_{Aj}=\Phi \quad (1)$$

$$E=(E'_A-E'_E)\cup E_I \quad (2)$$

$$E'_A=\{(v_i, v_j)\ |v_i, v_j\in V, \exists\ V_{Ai}, V_{Aj}\in V_{Ai}, v_i\in V_{Ai}, v_j\in V_{Aj}, (V_{Ai}, V_{Aj})\in E_A\} \quad (3)$$

如图 5 中说明的一样，$E_A$ 是一个同簇化后的图 $G_A$ 生成的完全联合图的边的集合，簇化技术算法的可实现性，使得可以有效地抽取集合 $E_A$，例如，完全联合成份。

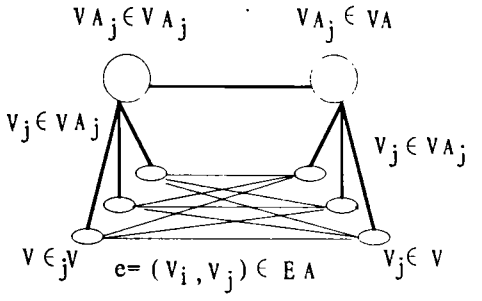


**图 5 $E'_A$ 在完全联合成份图中边的集合**

**Fig. 5 $E'_A$: A set of edges in the complete bipartite graph**

为达到最有效的簇化技术过程，希望达到最小化。

Cost—func ($G_A$, $G_I$, $G_E$)

$$=|V_A|+|E_A|+|E_I|+|E_E|\ k \rightarrow Min \qquad (4)$$

图 6 (a) 是以二进制写成的相邻矩阵的一个样本输入数据。矩阵的大小等于顶点数量。而元素 $X_{ij}=1$ 表示顶点 $X_i$ 和顶点 $X_j$ 存在一条边，0 则表示没有边，通过对最小开销函数使用，我们可以得到如图 6 (b) 的簇化矩阵。注意，0# 和 I* 分别表示专用链和共享链，这些链是例外的，也就是描述次要关系。

|V|=6

|E|=18

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 | 1 |
| 3 | 1 | 0 | 1 | 1 | 1 | 0 |
| 4 | 1 | 0 | 1 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 |

(a)

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 0# | 0 | 0 | 0 | 1 |
| 2 | 1 | 1 | 1 | 0 | 1* | 0 |
| 3 | 1 | 1 | 1 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1* | 1 | 0# | 1 |
| 5 | 0 | 0 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1* | 0 | 1 | 1 | 0# |

$|V_A|=2$

$|E_A|=2$

$|E_I|=3$

$|E_E|=3$

(b)

**图 6 ACE 的一个例子**

**Fig. 6 An example of ACE**

簇化技术算法是一种给定关系的簇化方法。允许在关系代数中设置差异和联合操作，由于簇化关系是一种在输入图中对结点属性的抽取。簇化技术算法可以当作将关系信息方法转换成结点的某些属性值对待。

## 4 簇化技术的分析模型

在这节中，将介绍一个簇化技术算法的分析模型，以便估量其中几个参数对有效的簇化技术的影响程度，以及将链减少以后所能减少的存贮空间。

当考虑在主存中物理数据库的设计时，存贮要求是很重要的，只有将超文本数据库进行简化后，才能有效地解决导航大型的超文本信息空间所出现的问题。作为这里讨论的目的，我们将提出如下的简化假设：

- 在每个簇的结点的数量一样，那就是，输入结点均匀分配到同等大小的簇中去。
- 专用链在相应的专用分区中的概率与共享链在相应的共用分区的。

下面的参数用于特征化这种模型

n——在输入图中的结点数量

p——输入链与可能的链的数量比值（因子）

比如 $pn^2$ 为输入图中的链数

q——在相应区域中例外链的平均因子

s——在簇化技术图 $G_A$ 中的结点的数量，也就是 $|V_A|$

r——簇化技术图的链数

α—是 s 与 n 的比值，即 $=s/n$，即每个输出簇的输入结点的超出平均数的簇

由于簇化的目标是减少每个分区中例外的比率，这里提出以下条件

$$0 \leqslant q < p \leqslant 1 \qquad (5)$$

由于输入链的总数等于组成簇化技术链加上共用链的数量相加之和，下面的条件是满足的

$$pn^2 = r\ (1-q)\ (\frac{n}{s})^2 + (s^2-r)\ q\ (\frac{n}{s})^2 \qquad (6)$$

因此

$$r = \frac{P-q}{1-2q} \cdot s^2 \qquad (7)$$

数据库的大小 S 取决于 n 以及簇关系的个数加上 $(r+qn^2)$ 以及在输出关系中的总数量。

因此

$$S = n + r + qn^2 \qquad (8)$$

为了估计最优化数据库的大小，必须考虑例外链的平均因子 q 与簇化技术图中结点数的关系 S，当 $S=1$ 时，q 等于 p，当 $S=n$ 时，输出图中没有例外链，$q=0$。因此，q 与 s 之间存在反比关系。

由于 q 受 $s^2$ 或 $\alpha^2$ 影响，如果链的分布是统一的，以及 n 是相对大时，可以提出以下条件。

$$q = p \cdot (1-\alpha^2) \qquad (9)$$

通过应用等式 (7) 和 (9)，以及存贮函数 α，等式 8 又可以写成

$$S = n + \frac{p\alpha^2}{1-2p\ (1-\alpha)^2} \cdot n^2\alpha^2 + pn^2\ (1-\alpha^2)$$

$$= n + pn^2 \cdot \left(\frac{(1-2p)\ \alpha^4 - (1-4p)\ a^2 + 1 - 2p}{2p\alpha^2 - 2p + 1}\right) \qquad (10)$$

对于条件 $\delta S/\delta\alpha=0$ 而言，α 的优化解法通过以下等式算出

$$\alpha^2_{opt}=\frac{\sqrt{1-2p}-(1-2p)}{2P} \quad (11)$$

也就是，$\alpha_{opt}$是P的函数，p是输入链的因子，与n无关，n是输入图中的结点数，如p的值远远小于1（$p<<1$），则$\sqrt{1-2p}\approx(1-p)$，并且$\alpha^2_{opt}$是近似0.5，当p趋向于0.5时，$\alpha^2$趋于零。

当输入图被当作较大的图看待时，例如$n<<pn^2$，并且$\sqrt{1-2p}\approx(1-p)$时，下面的优化条件得到满足

$$|E_I|_{opt}=\frac{1}{2}\cdot Pn^2 \quad (12)$$

$$|E_A|_{opt}+|E_E|_{opt}=\frac{1}{4}\cdot Pn^2 \quad (13)$$

由于输入数据库大小是$pn^2$，使用簇化技术算法将可以减小数据库大小的25%，图7根据$\alpha^2$给出了输出图数据库大小与输入图数据库大小之间的比较。当然，优化的结果取决于条件(9)，q与α的关系。如果输入图边（即链）相当密集，将得到更加密集输出图。反过来，如果输入图是随机的，减少的效果不大。

## 5 簇化技术的算法的描述

为了寻找一种簇化技术算法，簇化技术在Kernhan-lin算法[4]基础上发展一种启示性算法，Kernhan-lin是一种著名的图形分区算法，一个有加权的输入数据，具有偶数结点的无方向的图，可分成两个大小相等的结点集，也就是，在两个集中连接结点的所有边权的和变成最小。

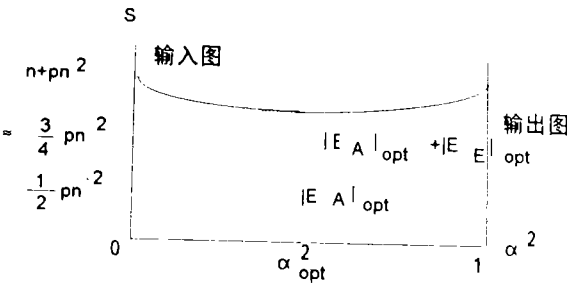


图7 数据库大小S与存贮函数α的比较

**Fig. 7 Comparison of the database size S with the storage function α**

通过修改Kernhan-lin算法以应用于簇化技术问题，这种改造算法中的差异是：组成每个簇结点的数量可能多于两个，并且，簇的大小不一定全都相等。这里的修改处在起始算法的每一步中使用一个虚构的簇。然后，重新安排簇，最终算法如下：

(Step1) Initial clustering step

Make all nodes belong to one cluster $C_1$ (thenumber of clusters: k=1)

(Step2) Repeated step

While no updating occurs do

Make $C_{k+1}$ as a dummy cluster

for i: =1 to n do (the number of nodes: n)

Choose some unselected node and call it vi

Let $j_c$ be the cluster of v (i. e., $vi\in C_{jc}$)

forj : =1 to k+1 and $j\neq j_c$ do

Calculate the cost when $v_i$ moves into $C_j$

Select the pair of $(v_i, C_j)$ if the movement makes the best benefit (i. e., largest decrease in cost)

end

Add $(v_i, C_j^{(1)})$ to the list of movement with the bestfit for this group of n×k alternative cases

end

Find l ($0\leqslant l\leqslant n$), s. t. $\sum Cost_l\rightarrow max$

Perform the translation $(v_1, C_j^{(1)})$, $(v_2, C_j^{(2)}, \cdots, (v_l, C_j^{(1)})$; that is, move vi into cluster $C_j^{(1)}$, (i=1, 2, …, l)

Rearrange clusters (k may change to (k−1) or (k+1))

end

## 6 讨论

输入结点和输出结点之间的簇关系被限制为一个对多对一的关系。这种限制允许关系的大小等于输入结点的数量n，其得出结果在简单输出结构中的簇里面，然而在某些特殊情况下，一些扩展方法提供更好的表达式。一种就是允许重叠簇，也就是在构造成簇的结点和它们的元素之间使用多对多的关系。当某些输入结点在两个簇集中关联时，它能提供一个压缩的簇化技术。另外一种扩展就是提供递归式，也就是，簇化后的图和例外图的有序性通过应用簇化技术算法递归地抽取。当同样的归并结构被用于在递归级中多级簇化技术时，它是很有用的。

另外一种扩展结果，例如，随机逼近的结合、加权链的考虑、簇化技术算法花费函数的定量评价、复杂度，都留到以后研究。

簇化算法的问题类似于图形划分问题，Johnson已经评价它们中的一些，并进行了精确的评价。已构成的可以是一种可选的实现算法，并产生靠近优化的结果。尽管它比递推的算法需要更多的时间。另外一种算法是基于贪婪算法，它非常快，但它却不能找到好的结果。因此，以后将在超文本结构中为生成浏览示意图时还要更多地应用这种算法。

## 7 结论

在这篇文章中，我们介绍了使用簇化原理从一个给定的超文本结构中获得高水平的关系结构。基于Kernighan-Lin算法，通过使用启发性的算法，利用数学模型对物理数据库问题的进行了评价。这种技术还使用了分析模型进行了优化分析。这种技术的结果表明了使用簇化方法对物理存贮上减少的用途。更潜在的用途是用于浏览和导航环境以及更符合于人类的界面设计。

## 参考文献


1 Yoshinori Hara et al. Implementing Hypertext Database Relationship Through Aggregations and Exceptions, Hypertext '91, 1991, 75: 15～18.

2 Conklin, Hypertext J. An Introduction and Survey, IEEE Computer. 1987, 20 (9): 17～41.

3 Hara Y, Kasahara Y. A Set-to-Set Linking Strategy for Hypertext Systems, ACM Conf. on OIS, 1990, 131～135.

4 Kenighan B W, Lin S. An Afficont Heuristic Procedure For Partitioning Graphs, Bell Sys J., 1970, 49 (2): 291～307.