基于 MapReduce 模型的分布式天文交叉证认 *

赵 青[1], 孙济洲[1], 肖 健[1†], 于 策[1], 崔辰州[2], 刘 旭[1], 袁 鳌[1]
(1. 天津大学 计算机科学与技术学院, 天津 300072; 2. 中国科学院 国家天文台, 北京 100012)

摘 要: 交叉证认是实现多波段数据融合的关键技术,目前还缺乏对其分布式算法的研究。快速增长的数据规模使该问题必须要依赖分布式并行计算技术解决。提出了一种基于 MapReduce 分布式模型的新方法,根据 MapReduce的要点,尽量减少了任务间的通信量,并通过合理设置划分粒度保证了效率与存储间的平衡。实验结果表明,该方法对海量数据交叉证认的效率提升明显,在大规模集群上达到了接近线性的加速比。该方法为交叉证认提供了一种快速有效的解决途径。

关键词: 天文交叉证认; MapReduce; 并行计算; 分布式计算
中图分类号: TP302 **文献标志码**: A **文章编号**: 1001-3695(2010)09-3322-04
doi:10.3969/j.issn.1001-3695.2010.09.032

Distributed astronomical cross-match based on MapReduce model

ZHAO Qing[1], SUN Ji-zhou[1], XIAO Jian[1†], YU Ce[1], CUI Chen-zhou[2], LIU Xu[1], YUAN Ao[1]
(1. *School of Computer Science & Technology, Tianjin University, Tianjin* 300072, *China*; 2. *National Astronomical Observatories, Chinese Academy of Sciences, Beijing* 100012, *China*)

Abstract: Cross-match is the kernel technology to realize multi-band data aggregation. It still remains blank in the research of its distributed processing functions. As the astronomical data is growing geometrically, it is inevitable to use distributed computing technologies to resolve it. This paper issued a new function based on MapReduce distributed computing model. According to MapReduce's design essentials, reduced the intra-node communication as far as possible, and insured a balance between efficiency and storage through choosing right partition granularity. The experimental results show that this function has a marked performance superiority comparing with previous functions, and achieves near-linear speedup in large-scale clusters. This new function is a quick and effective solution to astronomical cross-match problem.

Key words: astronomical cross-match; MapReduce; parallel computing; distributed computing

随着观测设备和技术的发展,天文学进入了全波段时代。不同的天文观测中心或天文设备针对天体不同的波段进行数据采集,将不同波段的星表进行数据融合以得到蕴涵更多信息的多波段或全波段数据,是天文学研究的基础,尤其对多波段数据挖掘和统计分析工作具有重要意义。所谓融合,其核心是基于不同星表的天体在位置、亮度等属性上的相关性,来确定某个星表中的某条记录与其他星表中的哪条记录为同一天体的过程,即交叉证认。交叉证认是多源数据联合查询的基础,同时也是望远镜选源阶段进行星体种类辨别的基础。近年来望远镜技术的飞速发展使各个研究机构所采集的数据量呈级数增长,对于如此规模的数据进行存储、交叉证认计算、检索等都不得不依靠于分布式环境。因此,针对此问题,本文研究设计了一种基于分布式环境的大规模交叉证认方法。

1 天文交叉证认及 MapReduce 模型

近年来各国的计算机专家都在进行交叉证认这一棘手问题的积极研究。图灵奖获得者吉姆· 格雷(Jim Gray)曾是美国虚拟天文台负责这一问题的首席科学家,他最早提出解决交叉证认问题必须要依靠并行计算技术。他以自己在数据库方面的深厚造诣,设计了内置于微软 SQL Server 的纯 SQL 指令的交叉证认算法[1,2],从而为大型天文巡天项目 SDSS 的数据访问平台整合了其他多家天文台的数据集。但这种方法在技术上受限于特定的数据库系统,证认的规模也受限于内存的容量,且规定一次证认的条数最多不能超过 5 000 条。虚拟天文台平台功能最为完备的英国虚拟天文台也在其网站的数据查询中提供了简单的交叉证认服务[3,4],但在数据规模和效率上仍然没有大的突破;其他国外开发的常用的交叉证认工具还有很多,如 Topcat、SDSS CasJobs、VizieR、Aladin 等,然而这些工具也只限于小规模数据上的交叉证认,而且也未考虑对分布式并行计算环境的支持。

近年来,我国对这一问题的研究需求也越来越迫切。刚刚落成的国家重大科学工程项目 LAMOST 望远镜以其超千万条的一夜观测光谱量成为了世界上光谱获取率最高的天文望远镜。按照天文研究的惯例,其他手段搜集到的资料要通过光谱

收稿日期: 2010-01-21; **修回日期**: 2010-04-01 **基金项目**: 国家自然科学基金资助项目(10978016);天津自然科学基金资助项目(08JCZDJC19700);天津市科技支撑重点项目(09ZCKFGX00400)

作者简介:赵青(1983-),女,天津人,博士,主要研究方向为并行计算、分布式计算;孙济洲(1949-),男,天津人,教授,博导,主要研究方向为分布式并行计算、计算机图形学;肖健(1978-),男(通信作者),河北人,助理工程师,主要研究方向为分布式计算(xiaojian@tju.edu.cn);于策(1979-),男,河北人,讲师,博士,主要研究方向为分布式并行计算、网格计算;崔辰州(1976-),男,河北人,副研究员,主要研究方向为虚拟天文台、天文信息技术;刘旭(1988-),女,本科生,主要研究方向为并行计算;袁鳌(1989-),男,本科生,主要研究方向为并行计算.

来确认,这一确认过程其核心就是交叉证认,可见多波段交叉证认关系到 LAMOST 数据能否被世界天文界所享用。在这样的背景下,我国在高效交叉证认方面也取得了一定的成果。高丹等人[5,6]提出了一种基于 HTM 球面索引和 kd-tree 的快速交叉证认算法,一定程度上满足了当时国内天文查询服务的需求,但该方法没有考虑分块后的边缘数据问题,漏源现象在所难免,而且其效率也只适于几十万条到几百万条的中等数据量。之后笔者在前人研究的基础上也提出了基于 HEALPix 球面索引和 MPI 的多核环境下的并行交叉证认方法[7],既解决了边缘数据问题导致的漏源现象,也使得几亿条数据量的证认可以在十几分钟内完成,一定程度上实现了海量数据的交叉证认。但这种方法仍然具有一定的局限性,一方面,要将今后 TB 级甚至 PB 级的海量数据存储于多核单机环境下并不现实,另一方面当前的证认速度也还与最终实现天文数据的实时查询的最终目标相差很远,因此迫切需要研究分布式环境下的大规模交叉证认高效方法。

MapReduce 是 Google 提出的一个并行框架,是 Google 搜索技术三大核心技术之一。这一模型非常适合处理 TB 级乃至 PB 级的海量数据,尤其适合待处理数据集可以分解成可以单独并行处理的许多小的数据集的情况。这与交叉证认的情景非常相似。a)交叉证认是典型的数据密集型计算,TB 级、PB 级的快速交叉证认服务对各国天文界来说都是迫切需要的;b)天文事业的非盈利性使得它对计算设备、存储设备的成本控制要求很高,而 MapReduce 模型正是一个可以充分利用大量低成本计算资源以获得高效率的模型。更重要的是,经过本文对交叉证认处理过程的改进,更可以保证证认阶段的子任务间的高度独立性,从而很好地利用了 MapReduce 的特性来最大限度地获得计算的高效性、规模的可扩展性及优异的加速比。因此,本文对基于 MapReduce 模型的交叉证认方法的研究是突破先前方法性能瓶颈的一种新的尝试,是可望实现实时交叉证认服务的一种新途径。

2　基于 MapReduce 模型的交叉证认

本文采用的交叉证认算法是基于位置信息的交叉证认,即对不同星表的每两条数据计算出它们之间的球面距离,当距离小于某阈值时就认为它们有可能是同一天体。这也是当今实现规模化数据融合的通用方法,美国虚拟天文台、英国虚拟天文台也都采用了这一方法。根据 MapReduce 模型的特点和适用范围,实现高效的交叉证认算法需要遵循以下原则:要将全部天文数据记录中的用于证认的信息字段(包括赤经、赤纬位置信息、星体 ID、索引等信息)分成大量的小块,并且要设法保证绝大多数的证认计算可以在单独的块内完成,从而尽量避免证认计算过程中节点间的通信。本章将根据这一原则针对四个方面来阐述基于 MapReduce 模型的交叉证认的具体方法。

2.1　数据划分方式及 HEALPix 索引算法

实现并行算法的第一步是对待处理数据进行分块,根据交叉证认的计算规则,只有在一定区域范围内的两个点才有可能为同一星体,所以按照位置对数据进行划分是实现数据分块、任务分配的最自然的方法,与此同时也实现了计算复杂度的降低。需进行交叉证认的范围一般按照式(1)(2)进行确定。其中 RA 和 DEC 分别为两个星表中的天体的赤经和赤纬坐标,r_1 和 r_2 为两个星表的观测误差半径。

$$|RA_A - RA_B| < (|r_1| + |r_2|)/\cos((DEC_A + DEC_B)/2)) \quad (1)$$

$$|DEC_A - DEC_B| < |r_1| + |r_2| \quad (2)$$

本文采取了一种天文上常用的球面索引方法 HEALPix,它以四叉树的递归方式对全天区进行等面积划分,如图 1 所示。它首先将全天区分成 12 个面积相等的球面四边形,然后在后续的每级划分中,每个四边形又被划分为四个相等的四边形子块。在编码上,也是采用逐级递归的方式,即父块的编码被继承为其子块编码的前缀,然后再根据每个子块的位置,分别在此编码的后面加上“00”“01”“10”或者“11”就构成了各个子块的编码。本文采用 HEALPix 进行数据划分的好处是:

a)其分块具有的等面积性避免了赤经坐标的额外修正。

b)HEALPix 已被广泛应用于全球范围内的天文数据访问服务中,基于它的交叉证认方法更具可扩展性,今后可与其他各种天文数据服务相融合。

数据切分过程中的另一个问题是边缘数据问题,即由于望远镜误差的存在,落于块的边缘上的数据,其在另一个星表中的对应体很有可能会落入另一个相邻的块内。边缘数据问题在各种方式的交叉证认中均存在,本文的方法也必须要解决这一问题。下面两节将具体阐述这一问题在 MapReduce 分布式算法中的解决办法及由此而产生的存储量与计算效率间的平衡问题。

2.2　边缘数据问题的解决方式和加速比的保障

如上文所述,对于块的边缘上的数据,其在另一个星表中的对应体很有可能会落入另一个相邻的块,所以在进行证认时,对于星表 A 中的一个计算块,在星表 B 中应选取同块数据以及其周围一圈的索引小块的数据与之证认。

由于分布式计算环境中节点间的通信比较耗时,应该尽量提高证认阶段各任务的独立性,减少数据的传输,这也是基于 MapReduce 的分布式计算的一个设计要点,同时也是保证算法在几十、几百节点的大规模集群上具有良好加速比的必要条件。本文采用打标记和复制数据的方法来解决边缘数据问题,以数据存储量的增加换来了证认计算中通信量的降低。具体来说,在预处理阶段对边缘数据进行多次拷贝并标记不同的块号,如图 2 所示,在星表 B 中,对于编码为“11000110”的数据块的相邻四条边上的数据,会分别被拷贝为两个副本,并分别打上它本来所属计算块的编码及“11000110”编码。这样,在 MapReduce 的交叉证认中,这些边缘数据既会和星表 A 中它原所属块的数据进行证认计算,也会和编码为“11000110”的块内的数据进行证认计算。而对于四个角上的数据则会被复制四次,并分别打上不同的标记后传到各个目的节点。

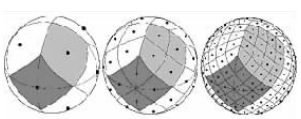

图1　HEALPix球面索引划分方式

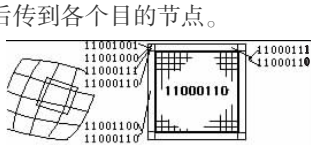

图2　边缘数据问题及打标记处理方法

2.3　计算块划分粒度的确定——效率与存储间的平衡

计算块分块数量决定了球面距离计算量和为了解决边缘数据问题所需副本数据的额外存储量。本节将从理论上针对如何在存储允许的条件下使计算效率达到最优的问题进行分

析,并在第 3 章中给出在各个分块粒度下的实验结果。

在数据访问服务中,HEALPix 索引的级数一般选为 13,本文也选取了这个级数,即全天区被划分为 12×4^{13} 个索引小块。经过计算,此时每一个索引小块的大小已经略大于式(1)(2)确定的需证认范围,所以程序中计算块的划分数量 12×4^{m} 中的 m 可以取值小于或等于 13 的任意整数。星表 A 和 B 中同属于同一计算块的两两记录间要进行一次球面距离计算,如果设两个索引小块间的平均距离计算量为单位 1,则全天区的球面距离计算量可以推导如下:

$$\begin{aligned}&\text{总球面距离计算量}=\text{计算块块数}\times\text{每块计算量}=\\&12\times4^{m}\times[4^{13-m}\times(2^{13-m}+2)^{2}]=\\&12\times4^{13}\times(4^{13-m}+4\times2^{13-m}+4)\end{aligned}\tag{3}$$

其中:m 为计算块的分块级数,因此每个计算块包含 4^{13-m} 个 HEALPix 索引小块。从式(3)可以得出结论:球面距离计算总量随着计算块总数的增加而减少。

效率问题不仅由球面距离计算量决定,根据交叉证认问题的各方文献,另一个主要性能瓶颈在数据 I/O 读取方面。经过实验测定,本文方法中,I/O 的耗时也大概占到了总耗时的一半左右,所以 I/O 操作的耗时会一定程度上削弱球面距离计算部分对整个程序运行效率的影响力。

此外,计算块划分过细将直接导致边缘数据所占比例的大幅上升,存储这些边缘数据的副本数据所需的额外存储量因而增加。例如,如果将计算块的分块粒度最大化,即以每个 HEALPix 索引小块为计算单元,则每一个计算块周围的 8 个相邻索引块就是它的邻居边缘数据,则每个计算块都是其他 8 个计算块的相邻边缘块,因此星表的存储量将高达原始数据的 9 倍。不同计算块分块粒度下所需存储量的大小可以表示为

$$\frac{\text{星表 B 的所需数据存储量}}{\text{星表 B 原始数据量}}=\frac{(2^{13-m}+2)^{2}}{4^{13-m}}=1+2^{m-11}+4^{m-12}\tag{4}$$

由此可以看出,随着计算块的划分粒度的加大,存储量的变化趋势与计算量的变化趋势正好相反,如图 3 所示。其中,总球面距离计算量的描绘是把计算块划分粒度达到最大值时(即 12×4^{13})的计算量作为 1 的,其他粒度下的球面距离计算量表示的是与此时计算量的相对比值。

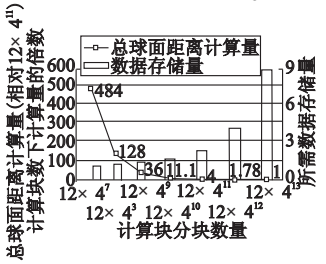

图3　计算量和数据存储量随计算块分块数的规律

如何选取最优分块粒度,使在不引入过多存储量的同时获得较好的性能还需要第 3 章中实验的实际测定。

2.4　MapReduce 算法的具体实现

为了使证认计算部分的效率达到最优,本文将整个交叉证认的计算过程分成了两个单独的 MapReduce 过程。第一个 MapReduce 过程实现了按照计算块块号对数据进行排序和向不同节点的分布式存放,所以在此之后,具有相同计算块块号的数据已存储于同一计算节点上。这一过程在构建交叉证认服务软件时,只在星表数据加入之时执行一次。第二个 MapReduce 实现的是证认计算,由于彼此间需要进行距离计算的数据已经存储于同一计算节点上,实际上这一过程只有 Map,没有 Reduce。把能在 Map 中完成的任务尽可能地放于 Map 中完成以省掉 Reduce 中的排序分发,也是 MapReduce 程序设计中的一个基本原则。

本文的基于 MapReduce 的并行交叉证认算法如图 4 所示,具体而言,两个阶段分别完成了以下工作:

阶段 1　数据的分布式存放(包含 Map、Reduce 两过程)

a)MapReduce 系统自动切分输入数据,完成数据块分发;

b)分布式地执行 Map 过程,对输入文件的每行,以块号 + 代表星表来源的标志位为 key,将其他字段的信息以空格间隔作为 value,输出〈key, value〉元组;

c)按 key 值排序、分区,并分发给各个节点,使具有相同块号的记录连续存放,从而构成一个个计算块组,组内相同来源的数据也是连续存放的;

d)各节点执行 Reduce,统计各组基本信息,包括各组来自星表 A 和 B 各自的条数,作为该组的头文件输出。

阶段 2　证认计算(只包含一个 Map 过程)

各节点执行 Map,对每一个计算块组,先读入头文件,如果来自两个星表中一方的数据量为 0,则对该块数据不执行任何操作;否则,将两星表数据分别读入两个数组,计算两数组间的每两条数据间的距离,如果小于特定值则输出〈ID_in_星表 A + ID_in_星表 B, 距离〉元组。

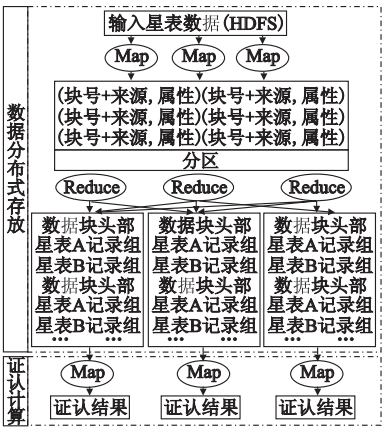

图4　基于MapReduce的交叉证认并行方法的执行过程

3　实验分析

本文采用 Hadoop 作为支持 MapReduce 程序的中间件,它是由 Lucene Apache 实现的 MapReduce 模型的开源软件框架。

实验环境是 64 台普通 PC,其中每台 PC 机配置如下:

CPU:奔腾双核 E2160,1800 MHz,内存:2 GB

操作系统:Linux Ubuntu 9.04; Swap:2 GB

编程环境:Java、Hadoop

测试数据是美国斯隆数字巡天 SDSS DR6 的星表 和 2 微米巡天计划的星表,其数据量分别约为 1 亿条和 4.7 亿条。

实验 1　最优分块数量的确定

本文 2.3 节已经对分块粒度与球面距离计算量间的关系进行了理论上的推导分析,图 5 中给出的是在 64 节点集群上证认计算部分在不同分块数下的性能测试结果,这部分的效率决定了实现实时交叉证认服务的可能性。

实验结果证明了 2.3 节中的两个分析:a)随着计算块块数的增多证认计算量逐渐减少,由此带来了证认总耗时的减少;b)I/O 读取的耗时、随着副本数量增多带来的额外数据处理的耗时会平抑距离计算量减少对整体效率的影响,所以总耗时曲线的下降幅度明显小于图 3 中球面距离计算量理论上的下降幅度。在存储量方面,实验结果与 2.3 节中理论上分析的结果基本一致。由此可知,分块粒度取到 12×4^{10} 或 12×4^{11} 比较合理,在没有引入过多存储量的同时,证认效率基本达到最高。

实验 2　随节点个数的增加证认效率加速比的测试

对于分布式程序,加速比的高低决定了算法能否具有良好的规模扩展性。本文将整个交叉证认过程分成数据的分布式存放和证认计算两个层次,从而将影响加速比提升的数据通信部分尽可能地解决在数据的分布式存放阶段,以求获得最好的证认计算性能。在 4、8、16、32、64 节点集群上、分块粒度下,证认计算部分的性能结果如图 6 所示。

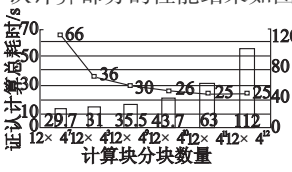

图5　证认计算耗时和数据存储量随分块数量的变化

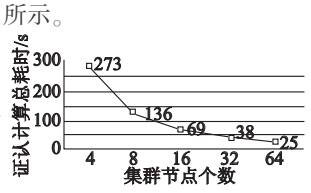

图6　不同节点个数下证认计算总耗时的变化情况

可以看出,从 4 ~ 32 节点,基本上达到了线性加速,即节点数每扩大一倍,时间约减少一半;而当节点数增加到 64 时,效率提高 30%。这与 Hadoop 计算过程有一定的启动时间有关,当证认时间缩减到一定程度时,启动时间占有的比例相应提高,致整体加速比减少。接近线性的加速比使本文方法继续推广到更大数据集、更大集群规模成为了可能。考虑到用户在提交交叉证认请求时多是针对某一天区的,而非当前的全天区证认,所以本文方法的效率已经基本可以满足实时交叉证认服务的需求。

实验 3　数据的分布式存放部分性能测试

在构建交叉证认服务或多源联合查询服务时,只有证认计算部分是用户提交请求后的操作,故只有这部分关系到服务实时性的实现。数据的分布式存放可看做预处理过程,只在一个数据集新加入时执行一次,因此这部分的性能只要满足基本要求即可。图 7 给出的是这部分在不同节点数下的执行时间,可以看出其耗时也是完全可以接受的。

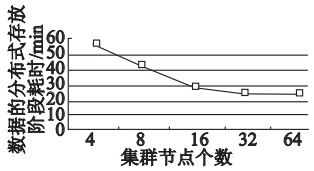

图7　证认数据的分布式存放部分耗时

实验 4　与其他方法性能对比

以上实验表明文中方法在上亿条的数据规模上确实表现出了很高的计算效率,25 s 的证认耗时相比先前多核环境下的并行方法[7]在完全相同数据集上的 32 min 的运行时间提升非常显著,而对比高丹等人[5]的方法的效率更是提高了万倍以上。

4　结束语

本文提出了一种基于 MapReduce 的天文交叉证认的分布式并行计算方法,将整个交叉证认过程分成了数据分发、证认计算两个独立的 MapReduce 过程。其中数据分发阶段是对数据的预处理,在交叉证认平台中只需对新加入的数据执行一次,所以设计上这部分尽量地包含了全部与通信相关的工作,从而使证认计算部分的各个子任务可以无须通信而独立完成,最大限度地利用了 MapReduce 模型的特性,保证了证认过程的高效性。实验证明,在上亿条的数据量级上此方法可以快速有效地完成交叉证认工作,其性能优于多核环境下基于数据库的并行交叉证认算法以及其他先前的方法,并且随着集群节点个数的增多,其性能表现出了接近于线性的加速比。可见,本文方法的提出为实现实时大规模交叉证认平台打下了基础,在提高天文学家对当今海量天文数据的利用效率方面起到了重要作用。

参考文献:

[1] GRAY J, SZALAY A, BUDAVRI T, *et al*. Cross-matching multiple spatial observations and dealing with missing data, MSR-TR-2006-175[R]. Redmond, WA: Microsoft Research, 2006.

[2] GRAY J, NIETO-SANTISTEBAN M A, SZALAY A S. The zones algorithm for finding points-near-a-point or cross-matching spatial datasets, MSR-TR-2006-52[R]. Redmond, WA: Microsoft Research, 2006.

[3] Report on cross matching catalogues, astroGrid[EB/OL]. (2007)[2008-11-09]. http://wiki.astrogrid.org/pub/Astrogrid/DataFederationandDataMining/cross.htm.

[4] Spatial joins and spatial indexing revisted, astroGrid[EB/OL]. (2007)[2008-11-10]. http://wiki.astrogrid.org/bin/view/Astrogrid/SpatialIndexing.

[5] 高丹,张彦霞,赵永恒. 海量多波段星表数据的交叉证认的实现[J]. 天文研究与技术, 国家天文台台刊, 2005, 2(3):186-193.

[6] 高丹. 海量天文数据融合系统的开发与数据挖掘算法的研究[D]. 北京:中国科学院国家天文台,2008

[7] ZHAO Qing, SUN Ji-zhou, YU Ce, *et al*. A paralleled large-scale astronomical cross-matching function[C]//Proc of the 9th International Conference on Algorithms and Architectures for Parallel Processing. Berlin: Springer, 2009:604-614.

(上接第 3315 页)

[15] 王三民. 模糊推理及态势估计研究[D]. 西安: 西安电子科技大学, 2004.

[16] 李弼程, 邵美珍, 黄洁,等. 模式识别原理与应用[M]. 西安: 西安电子科技大学出版社, 2008: 82-83.

[17] 网络舆情“智库”[EB/OL]. [2009-10-20]. http://www.trs.com.cn/news/gsxw/200910/tt20091020-2565.html.

[18] ATANASSOV K. Intuitionistic fuzzy sets[J]. Fuzzy Sets and Systems, 1986, 20(1):87-96.

[19] ATANASSOV K. More on intuitionistic fuzzy sets[J]. Fuzzy Sets and Systems, 1989, 33(1):37-46.

[20] 雷英杰,王宝树. 直觉模糊逻辑的语义算子研究[J]. 计算机科学, 2004, 31(11):4-6.

[21] 雷英杰,王宝树. 直觉模糊关系及其合成运算[J]. 系统工程理论与实践, 2005, 25(2):113-118.