第 43 卷 第 6A 期
2016 年 6 月

计 算 机 科 学
Computer Science

Vol. 43 No. 6A
June 2016

基于基本操作序列的编辑距离顺序验证

张润梁 牛之贤

(太原理工大学计算机科学与技术学院 太原 030024)

摘 要 两字符串的编辑距离是从一个串转换到另一个串所需要的最少基本操作数。编辑距离广泛应用于字符串近似匹配、字符串相似连接等领域。动态规划法利用编辑距离矩阵来计算两个串的编辑距离,需要计算矩阵中的所有元素,时间效率低。改进的方法改变了矩阵中元素的计算次序,减少了需要比对的元素,但仍需要比对一半以上的元素,时间效率还有待提高。提出基于基本操作序列的编辑距离顺序验证方法。首先,分析了基本操作序列的可列性,给出了列举基本操作序列的方法。然后依次顺序验证基本操作数从小到大的基本操作序列直到某一序列通过验证,得到其编辑距离。在阈值为 2 的字符串近似搜索实验中发现,所提方法比动态规划类方法具有更高的效率。

关键词 近似串搜索,编辑距离,顺序验证,基本操作序列

中图法分类号 TP391 文献标识码 A

Sequential Verification Algorithm to Compute Edit Distance Based on Edit Operation Sequence

ZHANG Run-liang NIU Zhi-xian

(College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024)

Abstract The edit distance between two strings is the minimum number of edit operations required to transform one into another. The edit distance is widely used in approximate string match, string similarity joins and etc. Dynamic programming algorithm(DPA) uses an edit distance matrix to compute the edit distance between two strings, which needs to compute all the elements in the matrix and has poor time efficiency. The progressive method changes the calculation orders of the elements to reduce the calculation numbers, which still needs to compute half of the elements and whose time efficiency needs to be improved. In our paper, we proposed a sequential verification algorithm to compute the edit distance based on the edit operation sequence. First, we analyzed the enumerable nature of edit operation sequence and gave a way to enumerate the sequences. Then, we got the result though sequentially verifying the edit operation sequences ordered by its edit operation numbers until the verification being successful. Experiments on approximate string search with threshold 2 show that compared with the DPA, our method achieves high performance.

Keywords Approximate string search, Edit distance, Sequential verification, Edit operation sequence

编辑距离广泛应用于字符串近似搜索[1,5]、字符串相似连接[2,4]等领域。相较于海明距离、余弦距离、jaccard 距离,编辑距离能够精确度量两个字符串的相似程度[1-7]。编辑距离的计算效率非常重要。一般采用动态规划法[1]计算编辑距离,该方法构建一个编辑距离矩阵,然后计算矩阵中每一个元素的值。文献[1]通过改变矩阵中元素的计算次序减少了需要比对的元素,但仍需比对一半以上的元素。为提高编辑距离的计算效率,本文提出一种基于基本操作序列的编辑距离顺序验证方法,首先给出列举基本操作序列的方法;然后给出基本操作序列顺序验证的方法;最后结合两个串的长度差,给出基于基本操作序列顺序验证求编辑距离的方法。该方法在计算小编辑距离时具有非常高的效率。

本文第 1 节介绍了编辑距离的基本概念及用动态规划法计算编辑距离的相关内容;第 2 节介绍了提出的基于基本操作序列的编辑距离顺序验证方法;第 3 节为实验;最后总结全文。

1 预备知识

编辑距离是从一个串转化成另一个串所需要的最少基本操作数,基本操作包括插入、删除、替换操作。用 $ed(r,s)$ 来表示串 r 和 s 的编辑距离,例如 ed("seraji","sraijt")=2。计算编辑距离一般采用动态规划法。

1.1 动态规划法计算编辑距离

该算法首先构建一个编辑距离矩阵,然后按照动态规划法逐行求出矩阵每个位置的值,直到得到矩阵的最后一个元素为两个字符串的编辑距离。

有字符串 r 和 s,用 $|r|$、$|s|$ 表示 r 和 s 的长度,$r[m]$ 表示 r 的第 m 个元素,$s[n]$ 表示 s 的第 n 个元素,编辑距离矩阵 D 是一个 $(|r|+1)\times(|s|+1)$ 的矩阵,用 $ed(m,n)$ 表示 $r[1,m]$ 和 $s[1,n]$ 的编辑距离。r="seraji",s="srajit"时,$ed(r,s)=2$,r 和 s 得到的编辑距离矩阵如表 1 所列。

张润梁(1987—),男,硕士生,主要研究方向为智能信息检索、字符串近似搜索;牛之贤(1963—),女,副教授,硕士生导师,主要研究方向为软件理论与算法、数据挖掘、管理信息系统,E-mail:niuzx@163.com(通信作者)。

表 1　动态规划法的计算编辑距离

		0	1	2	3	4	5	6
			s	r	a	j	i	t
0		0	1	2	3	4	5	6
1	s	1	0	1	2	3	4	5
2	e	2	1	1	2	3	4	5
3	r	3	2	1	2	3	4	5
4	a	4	3	2	1	2	3	4
5	j	5	4	3	2	1	2	3
6	i	6	5	4	3	2	1	2

首先初始化编辑距离的第一行和第一列，有 $ed[0,0]=0, ed[m,0]=m, ed[0,n]=n$。

对于矩阵中的任一位置 (m,n)，有 $ed(m,n)=\min(D(m-1,n)+1, D(m,n-1)+1, D(m-1,n-1)+\delta)$，其中如果 $r[i]=s[j]$，则 $\delta=0$，否则 $\delta=1$。

依据上述公式按行依次求出矩阵中的所有元素，最后得到 $ed(|r|,|s|)$ 为 r 和 s 的编辑距离。

动态规划法的时间复杂度为 $O(|r||s|)$，空间复杂度为 $O(|r||s|)$，其花费大量时间去计算无关的文本区域。时间效率比较低。

文献[1]提出了一种新的策略，改变了矩阵中元素的计算顺序。用 (i,j) 定义矩阵中第 i 行第 j 列的元素，用 Ex 定义那些值为 x 的元素。如果 $(|r|,|s|)\in Ex$，那么 $ed(r,s)=x$。这样，在计算 $ed(r,s)$ 时，只需让 x 从 0 开始计算 Ex，如果 $(|r|,|s|)\in Ex$，那么 $ed(r,s)=0$，否则基于 E0 计算 E1，如此迭代地计算，直到 $(|r|,|s|)\in Ex$。最后得到其编辑距离为 x。表 2 列出了改进的动态规划法得到的编辑距离矩阵。该方法可以减少很多不必要的比对，进一步提高了效率。

表 2　改进的动态规划法

		0	1	2	3	4	5	6
			s	r	a	j	i	t
0		0	1	2				
1	s	1	0	1	2			
2	e	2	1	1	2			
3	r		2	1	2			
4	a			2	1	2		
5	j				2	1	2	
6	i					2	1	2

2　基于基本操作序列的编辑距离顺序验证

基本操作序列是由编辑距离基本操作构成的序列，这里的基本操作是有序的，基本操作数为其编辑距离。基本操作有插入(insert)、删除(delete)、替换(substitute)，本文分别用 i、d、s 表示。$n(\text{ins})$、$n(\text{del})$、$n(\text{sub})$ 分别表示某一种操作的数目。由于 $n(\text{del})$、$n(\text{sub})$ 会影响两个串的长度差 $|r|-|s|$（$|r|$、$|s|$ 分别表示 r 和 s 的长度），分析基本操作序列时，需要考虑其长度差。

例，当两字符串间的编辑距离为 1 时，它有且仅有以下 3 种情况：1）有一个 del 操作；2）有一个 ins 操作；3）有一个 sub 操作。有一个 del 时，其对应的长度差为 1，其基本序列可表示为 1d；有一个 sub 时，长度差为 0，其基本序列可表示为 0s；有一个 ins 时，长度差为 −1，其基本序列可表示为 −1i。通过改变两个串的顺序可令 $|r|-|s|\geqslant 0$，这样编辑距离为 1 时，就只有 1d 和 0s 这两种基本操作序列。

上面提到了编辑距离为 1 时的情况，下面分析更一般的情况。

2.1　列举基本操作序列

给定 τ，有 $|r|-|s|\leqslant\tau$，则 τ 值就确定了 r 和 s 基本操作的数量和可能的长度差，这个长度差是有限可列举的。对每一个确定的长度差下基本操作数的组合进行排列，可列举出所有的基本操作序列。

按照这个思路，得到了列举基本操作序列的方法。首先根据 τ 得到其可能的长度差，其次列出每一种长度差可能的基本操作组合，然后对这些基本操作组合作排列，最后得到所有的基本操作序列。

列出基本操作时，长度差和 τ 值对 $n(\text{del})$、$n(\text{ins})$、$n(\text{sub})$ 的组合有以下约束：

(1) $n(\text{del})+n(\text{ins})+n(\text{sub})=\tau$，基本操作数等于编辑距离。

(2) $|r|-|s|=x$ 时，$n(\text{del})-n(\text{ins})=x$，因为 $n(\text{ins})\geqslant 0$，所以 $n(\text{del})\geqslant x$，长度差决定 $n(\text{del})$ 的下界。当 $|r|-|s|=0$ 时，有 $n(\text{del})=n(\text{ins})$。

根据以上两个约束，对每一个确定的 τ 和 $|r|-|s|$，可以穷举 3 种操作可能的组合，然后结合其顺序关系就可以列出所有的基本操作序列。

编辑距离 τ 较小时，基本操作序列很少，列举很容易。随着 τ 的逐渐变大，基本操作序列越来越多，列出基本操作序列越来越困难。本文列举了 $\tau=0,1,2,3$ 时对应的 $|r|-|s|$ 以及所有的基本操作序列（$|r|-|s|\geqslant 0$），如表 3 所列。

表 3　编辑距离为 0−3 时的基本操作序列（$|r|-|s|\geqslant 0$）

id	τ	|r|−|s|	基本操作数			基本操作序列
			sub	del	ins	0/1/2/3：长度差，o/s/d/i：基本操作类型
1	0	0	0	0	0	0o
2	1	0	1	0	0	0s
3		1	0	1	0	1d
4	2	0	0	1	1	0di；0id
5		0	2	0	0	0ss
6		1	1	1	0	1sd；1ds
7		2	0	2	0	2dd
8	3	0	3	0	0	0sss
9		0	1	1	1	0sid；0sdi；0dis；0dsi；0ids；0isd
10		1	2	1	0	1ssd；1sds；1dss
11		1	0	2	1	1ddi；1did；1idd
12		2	1	2	0	2sdd；2dsd；2dds
13		3	0	3	0	3ddd

2.2　基本操作序列的顺序验证

基本操作序列顺序验证就是按照两个串的顺序比对两个串的对应位，如果两个串对应位不相等，按照基本操作序列对其处理。如果按照基本操作序列处理两个串能处理的所有对应位，则两个串通过验证。每一次验证都只需比较 $\max(|r|,|s|)$ 次。

具体地，对应位的处理如下：

(1) 如果 $r[i]=s[j]$，r 和 s 当前位没有操作，继续比对 r 和 s 下一位（即令 $i=i+1, j=j+1$，再次比对 $r[i]$ 和 $s[j]$）。

(2) 如果 $r[i]\ !=s[j]$，则 r 和 s 当前位有一个基本操作，这时按照基本操作序列选择。

1）如果为替换操作，则 $n(\text{sub})$ 加 1，比对 r 和 s 下一位（$i=i+1, j=j+1$）；

2）如果为增加操作，则 $n(\text{add})$ 加 1，比对 r 和 s 下一位（$j=j+1$，i 不变）；

3)如果为删除操作，则 $n(\mathrm{del})$ 加 1，比对 r 和 s 下一位（$i=i+1, j$ 不变）。

依此类推，如果按照某一基本操作序列去处理每一个不相等的对应位，r 和 s 都到达结尾，则该序列通过验证，表示 r 能通过这一基本操作序列转化到 s。

下面用基本操作序列顺序验证的方法求编辑距离。

2.3 基于基本操作序列的编辑距离顺序验证

2.2 节给出检测两个串是否能通过某一操作序列进行转化的方法。编辑距离是求两个串进行转化的最少操作数，所以基于基本操作序列顺序验证求编辑距离需要从编辑距离为 0 的序列开始验证，如果通过验证，两个串编辑距离为 0；不通过则需要按编辑距离为 1 的序列验证。如果通过验证，两串编辑距离为 1，不通过，则验证编辑距离为 2 的序列。以此类推，直到验证通过，得到两串的编辑距离。由于是从小到大来验证，第一次检测通过的序列就是其最小的编辑距离，因此基于基本操作序列验证的方法得到的结果是准确的。

具体操作时，先求其长度差，然后根据长度差找到其对应的基本操作序列，最后按位顺序检测。例如，$|r|-|s|=0$ 时，可能的编辑距离范围是 0 至 $|r|$，首先按照 0o 序列来验证，若符合，$ed(r,s)=0$，如果不符合，再按照 0s 序列来验证，如符合，$ed(r,s)=1$，如果不符合，再按照 0ss，0da，0ad 来验证，以此类推，直到找到那个可以验证通过的序列。这个序列对应的编辑距离就是 r 和 s 的编辑距离，具体如算法 1。

算法 1 基于基本操作序列验证求编辑距离

Input：string r and s，ensure $|r|\geqslant|s|$

Output：ed(r,s)

1. a=|r|−|s|；
2. for(t=0；t<=|r|；t++)
3. if(r 和 s 通过长度差为 a、编辑距离为 t 的一个基本操作序列验证)
4. retrun t；

进一步分析发现，某些序列可以结合在一起进行检测，比如 0o、0s 和 0ss 对应 r 和 s 之间有 0、1、2 个替换操作，可以一起检测。

从表 3 可知，r 和 s 的编辑距离变大时，需要验证的次数越来越多，耗费的时间也越来越多。不考虑序列合并检测的情况，当 $|r|-|s|=0, ed=1$，需要验证 2 次；$ed=2$，最多需要验证 5 次；$ed=3$ 时，最多需要验证 12 次；$ed=4$ 时，最多需要验证 42 次。这种方法只在编辑距离较小时的情况（编辑距离小于等于 3）具有较高的效率。

2.4 基于基本操作序列顺序检测的阈值为 2 的近似串搜索算法

阈值为 2 的字符串近似搜索就是在候选字符串集中找到与查询字符串编辑距离小于或等于 2 的所有字符。

基于基本操作序列的顺序验证的方法在验证编辑距离为 0、1、2 的字符串对时具有很高的效率，下面结合长度过滤来解决这个问题。

当编辑距离小于或等于 2 时，有 $|r-s|\leqslant 2$。

(1)当 $|r-s|>2$ 时，直接过滤。

(2)当 $|r-s|=2$ 时，可能的基本操作序列为 2dd，直接检测删除操作即可。

(3)当 $|r-s|=1$ 时，可能的基本操作序列为 1d，1sd，1ds。

(4)当 $|r-s|=0$ 时，可能的基本操作序列为 0o，0s，0ss，0da，0ad，其中 0o，0s，0ss 可以合并为一次检测，所以只需做替换检测，即 0da，0ad 检测。

综上得到阈值为 2 的近似串搜索算法。详见算法 2。

算法 2 基于基本操作序列的顺序检测法的阈值为 2 的近似串搜索算法

Input：S：A string set；s∈S，q：A query

Output：R：A set，∀r∈R，ed(r,q)<=2

1. for each s，ensure |s|>=|q|；
2. switch：a=|s|−|q|；
3. a=2：s 和 q 进行 2dd 检测，如果通过检测，将 s 加入 R；
4. a=1：s 和 q 进行 1d，1sd，1ds 检测；如果通过检测，将 s 加入 R；
5. a=0：s 和 q 进行替换检测，即 0da，0ad 检测。如果通过检测，将 s 加入 R；
6. return R；

3 实验

实验验证了基于基本操作序列顺序验证的方法在阈值为 2 的近似串搜索中的时间效率，同时验证了其在不同长度候选集中的时间效率。

3.1 实验环境及数据集

本文系统环境为 Intel Core i5 3.2GHZ 的 CPU，4GB 内存，1TB 的硬盘；系统为 CentOS6.5；实验工具为 GCC、CMAKE。

实验数据集有以下 3 个：

(1)Author 数据集：来自 DBLP 中 Author 数据，本例从中选取 4000000 条的人名记录将其作为候选字符串集。这些人名最长的为 11 个字符，最短的 2 个字符，平均长度 6.027。查询字符串集是从其中随机选取的 43 个字符串。

(2)Author1 数据集：按 author 中每 3 个串合成一个串的方法得到，候选串为 4000000 个，查询串是随机从 Author1 中选取的 43 个串。

(3)Author2 数据集：按照 author 中每 5 个串合成一个串的方法合成了 Author2。候选串为 4000000 个，查询串是随机从 Author2 中选取的 43 个串。

本文选取了以下几种算法作为对比算法：

(1)DPED：动态规划法，只计算编辑距离中值小于等于 2 的元素。

(2)LDPED：length 过滤+DPED，length 过滤会过滤掉所有与查询串长度差大于 2 的候选字符串。

(3)CDPED：content 过滤+DPED，content 过滤会比较 query 和 candi 中每个字符出现的次数，然后计算每个字符相差的次数，最后得到总的相差次数。如果总的相差次数大于 2，会被过滤掉。

(4)LCDPED：length 过滤+content 过滤+DPED。

(5)SDED，本文提出的基于基本操作序列的顺序检测法。

3.2 实验结果

首先在 author 数据集中比对以上 5 种算法的运行时间，运行时间结果如图 1 所示。

从图中可知，在已有的算法中 LCDPED 具有较好的效果，从 4000000 条记录中查找 43 个字符的运行时间为 430.5 ms，本文 SDED 算法运行时间为 300.3ms，相差 130.2ms。SDED 算法比其它 4 种算法的运行时间更短，即在这 5 种算法中 SDED 算法是最快的。

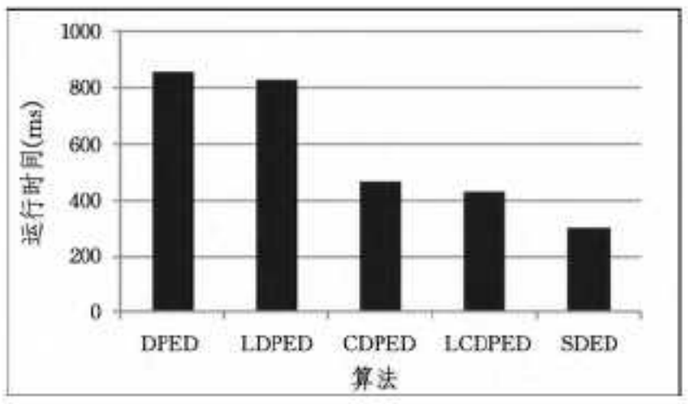

图 1　5 种算法的平均查询时间

下面对 LCDPED 和 SDED 算法在不同长度字符串中的性能进行比较，分别将 LCDPED 和 SDED 算法运用于 Author、Author1、Author2 中，得到的结果如图 2 所示。从图中可知，与 LCDPED 算法相比，本文算法不论在那个数据上都有显著的优势。

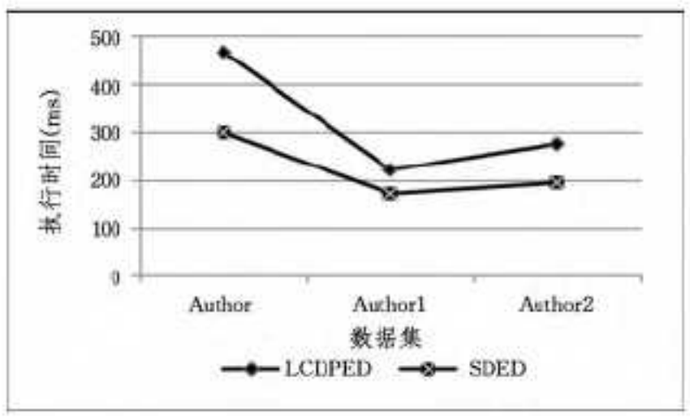

图 2　LCDPED 和 SDED 在不同长度字符串中的执行时间

结束语　本文提出了基于基本操作序列的编辑距离顺序验证方法。在阈值为 2 的近似串搜索实验中发现，相较于动态规划法及其改进算法，本文方法不论在长字符还是短字符中都具有良好的效果。

阈值较大时，本文方法需要验证的次数较多，时间效率会下降，所以本文算法适用于阈值较小的情况。

参 考 文 献

[1] Deng D, Li G, Feng J, et al. Top-k string similarity search with edit-distance constraints[C]// ICDE. 2013: 925-936

[2] Bayardo R, Ma Y, Srikant R. Scaling up all-pairs similarity search[C]// WWW Conference. 2007

[3] Chaudhuri S, Ganjam K, Ganti V, et al. Robust and Efficient Fuzzy Match for Online Data Cleaning[C]// SIGMOD. 2003: 313-324

[4] Jiang Y, Li G, Feng J, et al. String similarity joins: An experimental evaluation[J]. Proceedings of the VLDB Endowment, 2014, 7(8): 625-636

[5] Li C, Lu J, Lu Y. Efficient merging and filtering algorithms for approximate string searches[C]// ICDE. 2008: 257-266

[6] 纳瓦罗. 柔性字符串匹配[M]. 北京：电子工业出版社，2007

[7] 姜华，韩安琪，王美佳，等. 基于改进编辑距离的字符串相似度求解算法[J]. 计算机工程，2014，40(1)：222-227

[8] 黄亮，赵泽茂，梁兴开. 基于编辑距离的 Web 数据挖掘[J]. 计算机应用，2012，32(6)：1662-1665

[9] 薛晔伟，沈钧毅，张云. 一种编辑距离算法及其在网页搜索中的应用[J]. 西安交通大学学报，2008，42(12)：1450-1454

(上接第 39 页)

孤立性肺结节进行分类实验时，样本正例率(TPR)、反例率(TNR)和总体准确率都有了较大的提高，表现出了较好的分类性能，说明该方法在孤立性肺结节分类中是有效的。

表 1　不同分类方法的分类结果

方法	TPR(%)	TNR(%)	诊断正确率(%)
Bayesian hybrid classifier[2]	90.21	92.53	91.37
SVM[3]	87.39	91.29	89.34
BP neural networks[7]	84.80	90.34	87.57
LM-based BP networks[8]	88.02	91.82	89.92
Our method	93.33	95.25	94.29

结束语　在肺部影像的计算机辅助诊断系统中，孤立性肺结节的良恶性诊断是关键。为了提高诊断的准确性，本文提出了一种基于改进 BP 神经网络的肺结节分类算法，该算法基于孤立性肺结节的医学诊断属性，将遗传算法与 BP 神经网络结合起来，构造分类器，在准确特征提取的基础上实现了孤立性肺结节的良恶性分类。实验中分别采用文献中的方法、基本的 BP 神经网络以及本文方法对医院及网络公共数据集中的肺结节数据进行分类实验，根据分类结果验证方法的有效性。结果表明，优化后的算法表现出较好的分类性能，在分类准确性上较其它方法有了较大的提高，是肺结节临床分类方面的有效性算法。

参 考 文 献

[1] 强彦，卢军佐，赵涓涓，等. 基于 PET/CT 的孤立性肺结节的自动分割方法[J]. 清华大学学报(自然科学版)，2013，(2)：200-204

[2] Calle-Alonso F, Pérez C J, Arias-Nicolás J P, et al. Computer-aided diagnosis system: A Bayesian hybrid classification method [J]. Computer methods and Programs in Biomedicine, 2013, 112 (1): 104-113

[3] Keshani M, Azimifar Z, Tajeripour F, et al. Lung nodule segmentation and recognition using SVM classifier and active contour modeling: A complete intelligent system[J]. Computers in Biology and Medicine, 2013, 43(4): 287-300

[4] Tian G, Qi W. Target Recognition Algorithm Based on BP Networks and Invariant Moments[J]. TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013, 11(2): 969-974

[5] 李松，刘力军，刘颖鹏. 改进 PSO 优化 BP 神经网络的混沌时间序列预测[J]. 计算机工程与应用，2013，49(6)：245-248

[6] Kai S, Zhikun L, Hang S, et al. A research of maize disease image recognition of corn based on BP networks[C]// 2011 Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). IEEE, 2011, 1: 246-249

[7] 王先军，白国振，杨勇明. 复杂背景下 BP 神经网络的手势识别方法[J]. 计算机应用与软件，2013，30(3)：247-249

[8] D Shuo, C Xiao-heng, W Qing-hui, et al. Application of BP Networks Improved by LM Algorithm in Pattern Classification[J]. Electronic Test, 2014, 2: 16

[9] 彭基伟，吕文华，行鸿彦，等. 基于改进 GA-BP 神经网络的湿度传感器的温度补偿[J]. 仪器仪表学报，2013(1)：153-160

[10] 戴健伟，吉华，杨岗，等. 基于 GA-BP 算法的化工设备设计人工时预测[J]. 计算机集成制造系统，2013(7)：1665-1675

[11] 王平，王彩芸，王文健，等. GA-BP 网络在钢轨磨损量预测中的应用[J]. 润滑与密封，2011(2)：99-102，71