计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241000081

基于 BWT,MTF 和 ANS 的标签数据压缩算法

廖 睿 唐 杰 梁桐嘉 郑欣磊 王斌翊 齐志强
西北机电工程研究所 陕西 咸阳 712099
(3555878252@qq.com)

摘 要 使用一些规则集可以将一些信息转换为特定的内容存储在一定长度的标签码中。当信息较多时,标签码的使用会更加困难。对标签码数据进行压缩,可以减少存储信息的开销且便于识别。为了实现对此类数据的压缩,本文基于 BWT、MTF 和 ANS 算法,形成一种适用于标签数据的无损压缩算法,该算法在一定程度上对标签码进行无损数据压缩,有利于标签码信息的存储和识别以及标签码的使用。

关键词:标签数据;无损数据压缩;BWT 变换;前移变换;非对称数字系统
中图分类号 TP391

Label Data Compression Algorithms Based on BWT,MTF and ANS

LIAO Rui,TANG Jie,LIANG Tongjia,ZHENG Xinlei,WANG Binyi and QI Zhiqiang
Northwest Institute of Mechanical and Electrical Engineering,Xianyang,Shaanxi 712099,China

Abstract Using some rule sets it is possible to convert some information into specific content stored in a labeling code of a certain length. When there is a lot of information,it is more difficult to use the labeling code. Compression of tag code data can reduce the overhead of storing information and facilitate identification. In order to realize the compression of such data,this paper improves ANS based on BWT and MTF to form a lossless compression algorithm applicable to labeled data,which provides lossless data compression of labeled code to a certain extent,which is conducive to the storage and identification of labeled code information and promotes the use of labeled code.

Keywords Tag data,Lossless data compression,BWT transform,Move-to-front transform,Asymmetric Numeral Systems

在物资信息管理、物流跟踪、身份认证等领域,需要一种可以识别的标识信息用于区分个体和记录信息,这种包含一定数据量的标识信息称为标签码。在不同的使用场景中,标签数据会采用不同的编码规则,这使得标签码具有不同的表现形式。例如,在商品的包装上,常使用条形码或者二维码来表现商品的基础信息;在快递运输过程中,也有对应的标签码用于跟踪物流信息;在使用射频芯片记录信息的时候,标签码记录物品对应的基础信息存储在芯片中用于识别;还有《军用射频识别数据转换协议》(GJB7384-2011)用于记录军事物资信息。

以 GJB7384-2011 为例,物资信息经过特定的规则,转换为一串数据流,将这种转换后的信息存储在射频识别标签中,可以占用更少的存储空间[1]。基于 GJB7384-2011 的标签码可以用于物资的管理、装备的追踪和识别。通过使用符合该规则集的标签码,可以提高物资管理的效率和安全性,确保装备信息的准确记录和快速识别。有限的存储空间使得标签的识别处理流程要尽可能简化[2],如果可以用更少的空间存储更多的信息,就能降低标签识别的代价,提高识别速度。物资的运输、管理对效率和准确性的要求较高,不论是信息识别错误还是识别速度过慢,都是不能接受的。因此,越是简短的标签,就越有实际应用价值。

现有数据压缩的研究更多关注长字节的数据,对于短文本的标签数据研究较少,因此现有数据压缩方法主要关注多次重复出现信息带来的数据冗余,对短文本特征提取效果较差。因此,对于简短的标签数据,需要设计一种针对短文本的压缩算法实现数据压缩。

1 问题分析

标签数据是对物资信息特征的精炼表达,压缩标签数据的流程必须是可逆的,压缩编码和压缩解码的过程不能损失信息,因此对标签数据的压缩是一种无损压缩。在数据压缩领域,无损压缩有两种代表方法:基于统计的熵编码压缩和基于替换的字典压缩[3]。典型的字典编码压缩算法有 LZ77[4],LZW[5]等,熵编码压缩有算术编码[6]、非对称数字系统(ANS)[7]等。

在数据压缩中,能够被更短的内容替换的不定长子数据被称为模式,字典编码实现数据压缩的原理是使用更短的索引替换重复的模式,通过构建一个索引字典,将数据中重复出现的模式替换为字典中相应的索引。原始信息中重复的模式越多,替换的索引就越多,压缩效果就越好。短的标签数据可能有几十字节,长的标签可能有两千多字节,这样的数据中的重复模式会更少。在原始信息中重复内容较少的情况下,可能会导致不能统计出具有代表性的字典信息,使得压缩结果不理想。

熵编码旨在将数据中的符号用尽可能少的位数来表示,其核心思想是为更频繁出现的符号分配更短的编码,而为较少出现的符号分配更长的编码。其目的是最小化数据的平均表示长度,从而实现最优的压缩效果。相比于字典压缩算法,

通信作者:王斌翊(wangbyxx@outlook.com)

重复性低的内容对这种基于统计频率的算法的影响更小。

ANS作为一种兼顾压缩速度和压缩率的熵编码算法[8]，被广泛运用于许多领域。压缩数字图像中JPEG XL采用ANS就是它的一个成功的应用案例[9]。ANS主要分为3种：只接收二元输入的统一非对称二元系统(Uniform Asymmetric Binary System，uABS)、使用累积分布函数(CDF)确定每个符号编码的范围非对称数字系统(Range Asymmetric Numeral System，rANS)，以及构建编码表、将编码和解码过程列表的表非对称数字系统(Table Asymmetric Numeral System，tANS)。

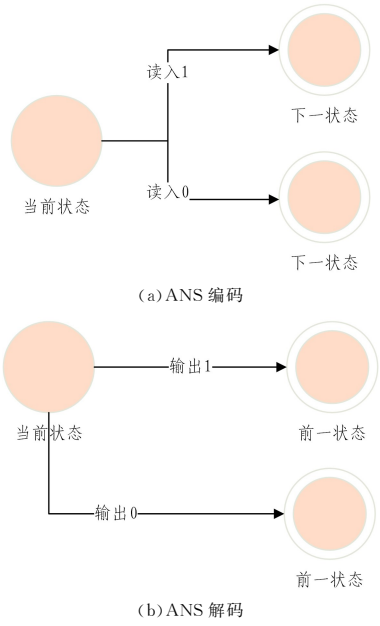


图1 ANS的状态转换形式

Fig. 1 State transition forms of ANS

ANS的核心思想就是频率越高的字符使用更少的位数表示。通过这种原则，可以对原本使用相同位数表达不同字符的原始字符串进行重新编码，以此实现数据压缩的目的。与算术编码一样，ANS也可以使用非整数位表达单个字符，但算术编码涉及对浮点数的乘除操作，这需要花费大量的时间，ANS以查找的方式替换浮点乘除法，使运行时间大幅度缩短。如果将ANS视为有限状态机，编码过程(见图1(a))和解码过程(见图1(b))可以表示为：

$$(input, currentstate) \rightarrow (nextstate) \tag{1}$$

$$(currentstate) \rightarrow (previousstate, output) \tag{2}$$

在上述两种主要压缩思想之外，数据压缩还包含对数据的预处理。不论是数据的重复模式，还是字符频率，其都是原始数据特征的一种表示，如果不改变原始数据内容，这种特征就不会发生变化。对于一些难以压缩的数据，如果不对其数据特征进行重新处理，则压缩算法执行的效果就不好。数据压缩领域有一些算法可以对原始数据进行预处理，例如bzip2就使用Burrows-Wheeler Transform算法(BWT)和Move to Front变换(MTF)对原始数据进行预处理[10]。此外，对于一些时序数据，有增量编码或小波变换等；针对连续出现的相同符号，可以使用游程编码进行预处理。

BWT算法最早由Burrows和Wheeler于1994年提出[11]。BWT的主要目的是对数据按照特定的规则进行重排列，使其中更多的相同的内容聚集在一起。BWT对数据的处理包含3个步骤：循环移位、排序和提取末尾字符串(L串)。对于给定的数据符号串“$ababc$”，首先向其末尾添加一个标志符号$，该符号可以是任意符号，在数据中视为比原始数据中任意符号小(针对于排序步骤)。然后对数据“$ababc\$$”循环移位，生成矩阵 $\boldsymbol{M}$：

$$\boldsymbol{M}=\begin{bmatrix} a & b & a & b & c & \$ \\ b & a & b & c & \$ & a \\ a & b & c & \$ & a & b \\ b & c & \$ & a & b & a \\ c & \$ & a & b & a & b \\ \$ & a & b & a & b & c \end{bmatrix}$$

循环移位生成 $\boldsymbol{M}$ 矩阵后，对矩阵的每一列数据串按照一定的规则进行排序(常选择字典序)，生成 $\boldsymbol{M}'$ 矩阵：

$$\boldsymbol{M}'=\begin{bmatrix} \$ & a & b & a & b & c \\ a & b & a & b & c & \$ \\ a & b & c & \$ & a & b \\ b & a & b & c & \$ & a \\ b & c & \$ & a & b & a \\ c & \$ & a & b & a & b \end{bmatrix}$$

$\boldsymbol{M}'$ 矩阵的第一列为字典序字符串(F串)，最后一列为L串，L串为BWT处理的最终结果。

BWT解码时不生成完整的二维矩阵，而是根据作为L串的结果“$c\$baab$”进行还原。首先对$L$串进行字典序排序，得到$F$串“$\$aabbc$”，在已知$F$串与$L$串后，可以得到原始数据内容。查找操作结束时，能够得到包含$的原数据逆序结果“$cbaba\$$”，将$去除并倒序，就是原始数据内容。还原原始数据的计算式为：

$$a_{\text{location}}=(A_{\text{first}}+B_{\text{Nth}}-1) \tag{3}$$

其中，A_{first}是上一个符号在F串中首次出现的位置；B_{Nth}是上一个符号在L串中对应位置上时，其已经出现的次数。此公式含义是每次读取字符，都通过上一个字符来查找下一个字符在L串中的位置a_{location}。

BWT变换通过循环排列原始数据来改变数据中的重复模式，将相同的字符更多地聚集在一起，增强数据的局部相似性[12]。这有助于统计模型更好地预测数据中的下一个符号，从而提高压缩效果。在执行BWT变换后，使用MTF变换可以进一步利用BWT变换生成结果的特性，帮助熵编码达到更好的压缩率。

MTF维持一个自适应的符号字典，该字典初始状态包含按升序排列的所有可能字符，该字典随着编码和解码的过程同步更新，并在编码解码结束时进行销毁，不需要单独保存：

$$D=\begin{cases} 0:a \\ 1:b \\ 2:c \\ \cdots \end{cases}$$

在依次读取符号串中的符号的过程中，每次读取到符号后，将该符号对应的字典项移动到字典的首位，如当读取到符号“c”时，更新后字典如下：

$$D'=\begin{cases} 0:c \\ 1:a \\ 2:b \\ \cdots \end{cases}$$

更新字典的同时使用字典中该符号索引对字符串中的该

字符进行替换。当读取到最后一位时，字典的任务结束，字典被销毁，将替换完成后的结果作为 MTF 的编码结果输出。

2　算法设计

与其他图像文件、时序文件不同，标签数据概括物体特征，它长度更短，在编码过程中对空间要求更小；它不是时序数据，不能使用增量编码减小数据的差值波动。因此，本文算法对标签数据的数据压缩使用 BWT 和 MTF 结合进行重排列、重编码，然后对重编码后的数据流进行压缩。

2.1　算法思想

本文算法分为 3 个步骤：使用 BWT 对原始标签数据进行重排序、再使用 MTF 变换进行重编码、最后使用基于累计分布函数的 ANS 进行无损压缩。

依据 GJB7384-2011 标准，原始标签数据包含汉字、字母、数字和符号等，对汉字采用 GB18030 进行编码，每个字母、数字、符号和汉字都有对应的编码，根据这些编码规则，可以将原始给定的数据转换为只包含 0 和 1 的比特流（见图 2）。

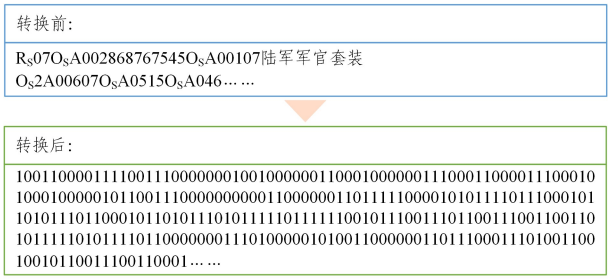


图 2　原始数据转换为二进制码示例

Fig. 2　Example of raw data converted to binary code

本文算法首先对原始标签数据对应的比特流进行压缩，再将其压缩结果中每 8bit 转储到单个字节中。本文在 BWT 处理时将其中每 4 bit 视为一个独立的符号进行整体 BWT，这样原数据中的所有内容就是由 16 种基础字符（见表 1）组成的一串二进制码。

表 1　基础字符

Table 1　Basic character

序号	符号	序号	符号
1	0000	9	1000
2	0001	10	1001
3	0010	11	1010
4	0011	12	1011
5	0100	13	1100
6	0101	14	1101
7	0110	15	1110
8	0111	16	1111

确定基础字符后，首先初始化一个空字符串 *res* 用于存放最终的编码结果。接着在标签数据字符串 *str* 末尾添加一个结束标识符“$ $ $ $”。然后对 *str* 进行循环移位，每次将字符串末尾字符（每 4 bit）移动到最前面，形成所有可能的字符串并记录在一个二维数组中。随后将这二维数组排序形成一个有序二维矩阵 $\boldsymbol{M}$。依次读取 $\boldsymbol{M}$ 的最后 4 列并拼接，形成包含结束标识符“$ $ $ $”的 L 串。然后使用固定 bit 位数记录标识符位置，将该位置信息加在 L 串头部用于替换原 L 串中无法用比特流表示的结束标识符“$ $ $ $”。

在上述 BWT 步骤完成后，使用 MTF 对其进行重编码。应用 MTF 时，为了便于利用前面 BWT 预处理数据的特征，本文算法将每 4 bit 视为一个独立的符号。首先创建一个初始字典，其中每项索引和其对应的符号相同（见表 2）。在遍历数据流时，每次都将遇到的 4 bit 在字典中的索引修改到第一项，其余索引项依次后移一位，使用修改后的新索引对读取的 4 bit 进行替换。例如数据中出现连续的一串 0101，即 010101010101，就将其编码为 010100000000，并在读取到第一个 0101 时将 MTF 字典中第一项修改为：{0000：0101}。

表 2　MTF 字典

Table 2　MTF dictionary

序号	索引	符号	序号	索引	符号
1	0000	0000	9	1000	1000
2	0001	0001	10	1001	1001
3	0010	0010	11	1010	1010
4	0011	0011	12	1011	1011
5	0000	0100	13	1100	1100
6	0001	0101	14	1101	1101
7	0010	0110	15	1110	1110
8	0011	0111	16	1111	1111

按照这种规则处理完成后，数据中的 0 和 1 频率会发生变化，本文算法使用 rANS 的思想，根据出现频率，为 0 和 1 分配不同的状态范围，编码函数 $C(s,x)$ 和解码函数 $D(x')$ 分别为：

$$C(s,x)=\left\lfloor\frac{x}{L_s}\right\rfloor\times L+CDF[s]+x \bmod L_s \tag{4}$$

$$D(x')=(x,s)=\begin{cases}x=\left\lfloor\dfrac{x'}{L}\right\rfloor\times L_s-CDF[s]+x' \bmod L\\ s=x' \bmod L\end{cases} \tag{5}$$

其中，L 是循环的长度；$CDF[s]$ 是累积距离函数，表示在编码表中，符号 s 之前所有符号的子周期长度之和。

在执行 rANS 压缩前，首先统计其中 0 和 1 的频率和数据流的长度，将其传给 rANS 编码算法，算法流程中的 Cs 变量即为累积距离函数值。最后存储时，压缩后的二进制码数据流可以每 8bit 存放在一个字节中，从而达到压缩的目的。

解码时，首先使用 rANS 的解码函数 $D(x')$ 循环计算前一状态，逐步还原压缩前的数据。然后将还原得到的比特流执行逆向 MTF 和 BWT 的逆变换操作，获得原始标签数据对应的比特流信息。

在 BWT 的逆变换操作中，首先需要初始化一个存放解码结果的空字符串 *res* 和位置指针 $location=0$。然而根据结束标识符位置信息，在对应位置处添加结束标识符“$ $ $ $”获得 L 串，最后对 L 串执行排序后得到 F 串，再迭代 L 串的每 4 bit 寻找字符，此时寻找字符的计算式为：

$$a_{\text{location}}=(A_{\text{first}}+B_{\text{Nth}}-1)*C_{\text{length}} \tag{6}$$

其中，A_{first} 是上一组 4 bit 在 F 串中首次出现时，第 1 bit 的位置；B_{Nth} 是上一组 4 bit 在 L 串中对应位置上时，其已经出现的次数，C_{length} 是基础字符占据的 bit 位数，此处 $C_{\text{length}}=4$。

2.2　算法实现

经过 BWT 和 MTF 的步骤，本文算法所用 rANS 在压缩完成时，除了生成的数据流之外，还含有最终状态 *Final state*、频率数组 *symbol_counts* 和原始数据长度 *length* 等附加信息，其中 *symbol_counts* 存储 0 和 1 的统计次数，就可以计算得出 *length*，这样附加信息就只包含频率数组 *symbol_*

counts 和最终状态 *Final state*，具体如算法 1 所示。

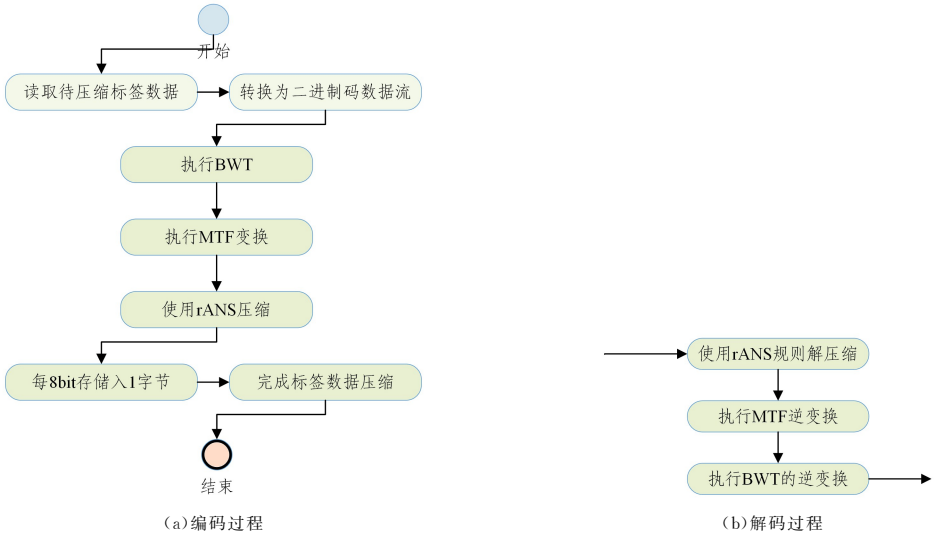
算法 1 rANS_coding

输入：待编码字符串 str

输出：编码结果字符串 res

1. 给待编码字符串 str 加入标识符 $ $ $ $
2. for i=0 到 str 的长度：
3. 获取当前 4bit 基础字符 ch
4. 将 ch 移动到 str 最前
5. 记录移动后生成的新字符串
6. 对循环移位生成的矩阵向量进行排序
7. 读取每一行最后 4bit 位生成处理结果 str2
8. for i=0 到 str2 的长度：
9. 获取当前 4bit 基础字符 ch
10. 在字典中查找 ch 的位置
11. 将索引加入 str3
12. 更新字典，将当前索引项移动到字典首位
13. 统计 str3 中 0 的频率 p_0，str3 的长度 length
14. 初始化 0 和 1 范围数组 cumul_counts
15. 初始化 0 和 1 频率数组 symbol_counts
16. 初始化状态变量 tempstate
17. 对于 str3 中的每 bit 字符 ch：//ch 为 0 或 1
18. 设置 s=ch
19. 设置 Fs 为 symbol_counts[s]
20. 设置 Cs 为 cumul_counts[s]
21. out_bits 初始化为空字符串，用于存储输出位
22. 当 tempstate 大于等于 2 * Fs 时：
23. 将 tempstate 取余数并加到 out_bits
24. tempstate / 2
25. tempstate =(tempstate / Fs) * length+Cs +(tempstate % Fs)
26. 将 out_bits 加到 res
27. 返回结果 res

程序实现压缩的流程为：使用 BWT 进行数据重排列、执行 MTF 变换利用 BWT 结果进行重编码、执行 rANS(见图 3(a))。解码则首先利用 rANS 的解码规则求出预处理数据，再对预处理数据依次执行逆向 MTF 和 BWT 的逆变换获得原始数据内容(见图 3(b))。



(a)编码过程　　(b)解码过程

图 3 算法流程图

Fig. 3 Flowchart of algorithm

3 实验分析

3.1 评估指标

压缩比是衡量数据压缩效率的一个关键指标，它反映了压缩算法在减小数据量方面的性能。其计算式为：

$$\text{压缩比}=\frac{\text{原始数据大小(bit 或字节)}}{\text{压缩结果大小(bit 或字节)}} \tag{7}$$

压缩比越高，说明压缩算法的性能越好。压缩比大小的影响因素有数据的冗余度、数据的随机性和压缩算法的选择等。数据和压缩算法都会影响压缩比结果：使用同一种压缩算法时，数据的冗余度越高，压缩比就越高；对于随机性较强的数据，数据间的关联性低，数据能够压缩的内容少，此时压缩比更低；不同的压缩算法适用于不同类型的数据，在选择不同压缩算法时，压缩比也通常会相应变化。

使用压缩比对压缩算法进行评估，既可以验证压缩算法的设计是否成功，也可以作为多种压缩算法之间的比较指标，分析不同压缩算法对数据的压缩效果。一个成功的无损压缩算法，对于待压缩数据的压缩比不小于 1；对于不同的压缩算法，压缩比越高，代表其压缩结果越短，说明该算法的压缩效果越好。

3.2 数据选择

标签数据的转换规则有许多种，不论哪一种，其用途都是为了方便标签数据的存储，GJB7384－2011(《军用射频识别数据转换协议》)规定了军用射频识别系统中数据转换的规则，适用于物品唯一标识和用户数据的转换。其规定的无源射频标签编码转换规则设定了标签数据中各专用标识符字段的数据类型和长度。本文实验依据 GJB7384－2011，生成不同长度的标签数据进行分析。

3.3 适用性

数据压缩算法应用广泛，数据内容本身特征会影响压缩效果，一串数据流，其中的重复信息越多，说明其数据冗余度越大，执行压缩算法后压缩比也越大。这种影响在字典压缩中，就是重复模式，可以用索引有效替换的重复模式越多，执行结果的压缩比就越大。在 rANS 这种熵编码算法中，则表现为字符的频率不均匀，单个字符的频率越高，就可以使用越

短的位数表达该字符。因此在验证数据压缩算法的有效性时，还要考虑它对待压缩数据的适用程度。

考虑一种极端例子：假设有一种物资，它的每一项特征都使用了不同的文本进行描述，此时生成的一段标签数据中不包含任何重复性词语、字母或数字。如果使用字典压缩就无法找到重复模式，但如果采用 rANS，尤其是从二进制编码流层面执行 rANS，那么就可以统计出其中二进制编码字符的频率不均匀性。

为了分析这种情况，本文使用国家统计的现代汉语常用字表中不重复的汉字作为标签数据进行压缩，计算得到其压缩比结果（见表 3）。

表 3　无重复汉字压缩率

Table 3　Compression rate of non-repeating Chinese characters

文本	文本长度/kB	压缩后长度	压缩比
常用汉字 1-833	2499	2123	1.1771
常用字 834-1667	2499	2209	1.1313
常用字 1668-2500	2499	2269	1.1014
平均值	2499	2200	1.1366

现代汉语常用字有 2500 个，每个汉字占据 2 字节或者 3 字节不等，总长度为 7500 字节，因此将其平均分为不超过 3kB 的 3 个部分进行压缩验证。表 3 中的结果表明，对于无重复的原始数据，本文算法依旧能达到一定的压缩效果，压缩后的文本长度约为原来的 88.04%。在实际标签数据中重复性内容会更多，所以这种结果表明本文算法是适用于标签数据的无损压缩的。

3.4　压缩比分析

使用式(7)对压缩结果的压缩比进行计算，得出不同长度的标签数据对应的压缩比（见表 4），以标签数据长度为横轴，绘制折线图（见图 4）。

表 4　本文方法和 LZW 算法的压缩结果

Table 4　Compression results of the proposed method and LZW algorithm

标签长度/kB	本文算法压缩比	LZW 压缩比
99	1.10	1.38
279	1.37	1.37
468	1.39	1.08
873	1.43	1.36
1222	1.57	1.63
2738	1.64	1.15

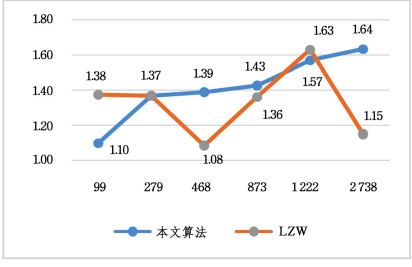


图 4　本文算法和 LZW 的压缩比

Fig. 4　Compression ratio of this paper's algorithm and LZW

由图 4 可知，字典编码 LZW 的压缩比在标签数据长度逐渐增长时，使用 LZW 的压缩比波动更大，尤其是当去除最短长度 99 字节时的压缩比数据时，本文算法趋势平稳，但 LZW 的压缩比依然在不断波动。对于最短长度的标签数据的压缩，本文所用的 rANS 压缩需要在压缩结果中存储固定长度的附加信息，这种附加信息记录最终状态、字符频率，用于解码时直接使用，而 LZW 算法所使用的字典信息是不需要存储的，它在编码和解码过程逐步推进时自动生成，压缩结果里不包含这类附加信息。受此影响，本文算法压缩比低于 LZW 算法。LZW 压缩比波动较大的原因是标签数据长度不足，通过字典压缩算法构建的自适应字典一定能很好地代表原始数据的数据特征，导致压缩率不稳定。若字典索引刚好可以对数据的重复模式进行有效替换，就会得到更高的压缩比；而当字典索引不断更新，却在原始数据中没有找到对应的重复模式，就会导致压缩比偏小。

本文算法的压缩效果主要由数据中字符频率决定。与字典压缩算法的自适应字典不同，本文算法对字符频率利用是基于标签数据整体的字符频率，它不取决于原始标签数据中的重复模式，因此不论是较短还是较长的标签数据，压缩比都较为稳定。

3.5　同类熵编码算法对比

与 LZW 算法的比较结果表明本文算法对标签数据具有无损压缩效果。这种压缩效果不仅仅源于压缩算法 rANS，BWT 和 MTF 的预处理步骤也不可或缺。为了验证预处理的有效性，将本文算法和 rANS 进行比较。

从表 4 中可以看出，rANS 压缩比小于完整执行预处理步骤的本文算法，且本文算法在加入预处理步骤后，压缩、解压速度波动不大。

表 5　压缩比结果

Table 5　Compression ratio results

文本长度	使用算法	编码结果长度	压缩比	压缩时长/ms	解压时长/ms
468	rANS	374	1.25	49	101
	本文算法	337	1.39	56	95
1222	rANS	805	1.52	113	213
	本文算法	777	1.57	126	220
2738	rANS	1752	1.56	298	498
	本文算法	1673	1.64	301	465

BWT 是对原始数据的重新排列，经过这种排列可以增强数据的局部相似性，MTF 利用这种性质，改变原有数据中各字符的频率，对 rANS 压缩算法起到正向的促进作用。BWT 和 MTF 对数据的操作不涉及浮点运算，因此在执行速度方面不会对压缩算法造成太大影响。

3.6　结果分析

实验分析结果表明，本文算法可以实现对标签数据的无损压缩。在执行 GJB7384－2011 编码转换后使用这种压缩算法对二进制编码进行处理，可以在一定程度上缩短原来的标签数据长度。这样在识别标签数据时，识别设备的压力会有所减小，识别效率就能够得到提高；同时在存储空间不变的情况下，执行这种压缩算法，可以避免在物资信息有所变动时难以加入更多的标签内容。这对于标签数据的使用有着积极的意义。

结束语　本文将 BWT，MTF 和 ANS 压缩算法相结合，形成一种适用于标签数据的无损压缩算法。实验结果表明，通过采用 BWT 和 MTF 的预处理方法，ANS 可以对标签数据进行无损压缩，平均压缩比为 1.44，可以使原有数据长度缩短 20%左右；无重复汉字的压缩比表明本文算法在压缩无重复汉字时也能使数据长度缩短 12%。本文压缩算法在标

签所含数据量相同时，识别所需读取的内容更少，提高了标签的识别效率；在存储花销相同时，所含数据会更多，有利于更多标签信息的存储。

当前对标签数据的压缩研究较少，本文研究为标签数据的压缩提供了一种解决思路，但更多是与压缩思路进行比较，使得评估结果具有一定的局限性。未来的研究可以考虑对预处理算法进行改进，使用更优良的预处理方式；也可以考虑通过应用深度学习找到更能够代表数据的特征信息，从而实现更好的压缩效果。

参 考 文 献

[1] GJB 7384-2011. 军用射频识别数据转换协议[S]. 2011-09-06.

[2] WANG Y. Research on High Efficiency and Robust Anti Collision Algorithm in RFID System [D]. Chengdu: Southwest Jiaotong University, 2019.

[3] PAREKAR P M, THAKARE S S. Lossless Data Compression Algorithm－A Review[J]. International Journal of Computer Science and Information Technologies, 2014, . 5(1): 276-281.

[4] RIGLER S, BISHOP W, KENNINGS A. FPGA-Based Lossless Data Compression using Huffman and LZ77 Algorithms[C]// 2007: 1235-1238.

[5] YE H N. Research on an Improved LZW-FSE Data Compression Algorithm[D]. Harbin: Harbin University of Science and Technology, 2021.

[6] LIU Y, JIANG L, LI Y C H, et al. Research on Dynamic Reconfigurable Implementation of Adaptive Binary Arithmetic Coding [J]. Electronic Measurement Technology, 2022, 45(19): 6.

[7] DUDA J. Asymmetric numeral systems as close to capacity low state entropy coders[J]. CoRR, 2013, abs/1311. 2540.

[8] CAMTEPE S, DUDA J, MAHBOUBI A, et al. Compcrypt-Lightweight ANS-Based Compre-ssion and Encryption[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 3859-3873.

[9] HSIE H, PING A, WU J L. A Review of the Asymmetric Numeral System and Its Applications to Digital Images. [J]. Entropy, 2022, 24(3): 375.

[10] SZECOWKA P M, MANDRYSZ T. Towards hardware implementation of bzip2 data compression algorithm[C]// 2009 MIXDES－16th International Conference Mixed Design of Integrated Circuits & Systems. 2009.

[11] BURROWS M, WHEELER D. A block-sorting lossless data compression algorithm[R]. SRC Research Report 124, Digital Systems Research Center, Palo Alto, CA, USA, 1994.

[12] LI B, LONG B J, LIU Y. A fast implementation of Burrows-Wheeler transform based on suffix ordering[J]. Journal of Electronics and Information, 2015, 37(2): 504-508.

LIAO Rui, born in 2001, master. His main research interests include intelligent sensing and control engineering of equipment.

WANG Binyi, born in 1975, master, research fellow. His main research interests include computer technology, artillery, automatic weapons and ammunition engineering.