Bit Recycling with Prefix Codes

Danny Dubé* Vincent Beaudoin†
Université Laval
Canada

Many data compression methods fail to remove all redundancy from a clear-text file partly because they allow it to be encoded into many different compressed files. The compressed files are *different* in the sense that they are different sequences of bits. However, they are *equivalent* in the sense that, by decompressing any of them, we recover the clear-text file exactly. The existence of multiple encodings tends to increase the size of the compressed files. One of the causes behind multiplicity is the existence of *equivalent messages*. Let us view a compressed file as a sequence of messages $M_1 \dots M_N$ transmitted from the compressor to the decompressor. At step i, M'_i is said to be equivalent to M_i if $M_1 \dots M_{i-1} M'_i M_{i+1} \dots M_N$ is equivalent to $M_1 \dots M_N$. In this work, we address redundancy caused by the existence of equivalent messages only. An obvious example where equivalent messages occur is the LZ77 family of compression methods where there may be more than one *longest* match, in which case all the longest matches are equivalent.

This paper presents a technique that aims at reducing the expansion of the compressed files that is caused by the multiplicity of equivalent messages. It does not try to eliminate multiplicity. Instead, it takes advantage of multiplicity by converting it into useful information, which we choose to describe parts of the compressed file itself. We call this technique *bit recycling*. On the decompressor side, when a message M is received, the set $\overline{M}$ of messages equivalent to M is determined, and the particular choice ($M \in \overline{M}$) made by the compressor is perceived as a hint, which translates into a bit sequence. Such a bit sequence is said to be *recycled* and the bits it contains can be omitted from the compressed file. On the compressor side, the task is more complicated because the message that is *currently* selected among the set of equivalent ones carries information about the *following* messages. To make these far-reaching selections, the compressor may use non-deterministic choices but we propose a *resolution* algorithm along with a greedy version that allows the compressor to proceed in a stream-like fashion. We propose two ways to obtain recycled bit sequences: *flat* recycling, where a constant number of bits (about $\log_2 |\overline{M}|$) is recovered for any selection of $M \in \overline{M}$; and *proportional* recycling, where the number of bits that is recovered for the selection of $M \in \overline{M}$ grows with the cost of encoding M. In a 2006 paper, Dubé and Beaudoin showed that they obtained the best experimental results using proportional recycling. We believe this recycling method to be close to optimal.

In previous work, the multiplicity of compressed files has been exploited to perform information hiding (and related applications). Multiplicity of equivalent messages has also been exploited for recycling purposes by Yokoo *et al.* In future work, recycling ought to be tuned to become optimal with prefix codes, adapted to arithmetic coding, and extended to exploit redundancy that is *not* caused by equivalent messages.

*Danny.Dube@ift.ulaval.ca †Vincent.Beaudoin.1@ulaval.ca