Multi-Objective LZSS Compression

Andrea Farruggia, Paolo Ferragina
Department of Computer Science, University of Pisa

A New Compression Paradigm

Modern applications demand more than just compression ratio: decompression time, energy consumption... conflicting, multiple objectives!
This issue is usually dealt with by "tweaking" heuristically a known compression method until it exhibits the desired trade-off.

Our aim: design a compression method which lets the user choose in a principled way the required trade-off.

The LZSS compression method

Compression is performed in two steps:

- *Parsing*: the text is decomposed in phrases (strings) belonging to the LZSS dictionary.
- *Encoding*: mapping copies and literals to *codewords*, i.e., binary strings of variable length.

Decompression is performed by iteratively *decoding* a codeword and *copying* the corresponding phrase, thus deriving the original phrase.

abaabaab
↓ *Parsing*
(0, a) (0, b) (0, a) (3, 5)
↓ *Encoding*
1a1b1a0001000001
↓ *Decoding*
(0, a) (0, b) (0, a) (3, 5)
↓ *Copying*
abaabaab

The parsing strategy and the encoding format can be customized and both greatly affect decompression times. In our case, the pattern of memory accesses induced by the parsing is fundamental in determining the decompression time/compression ratio trade-off.

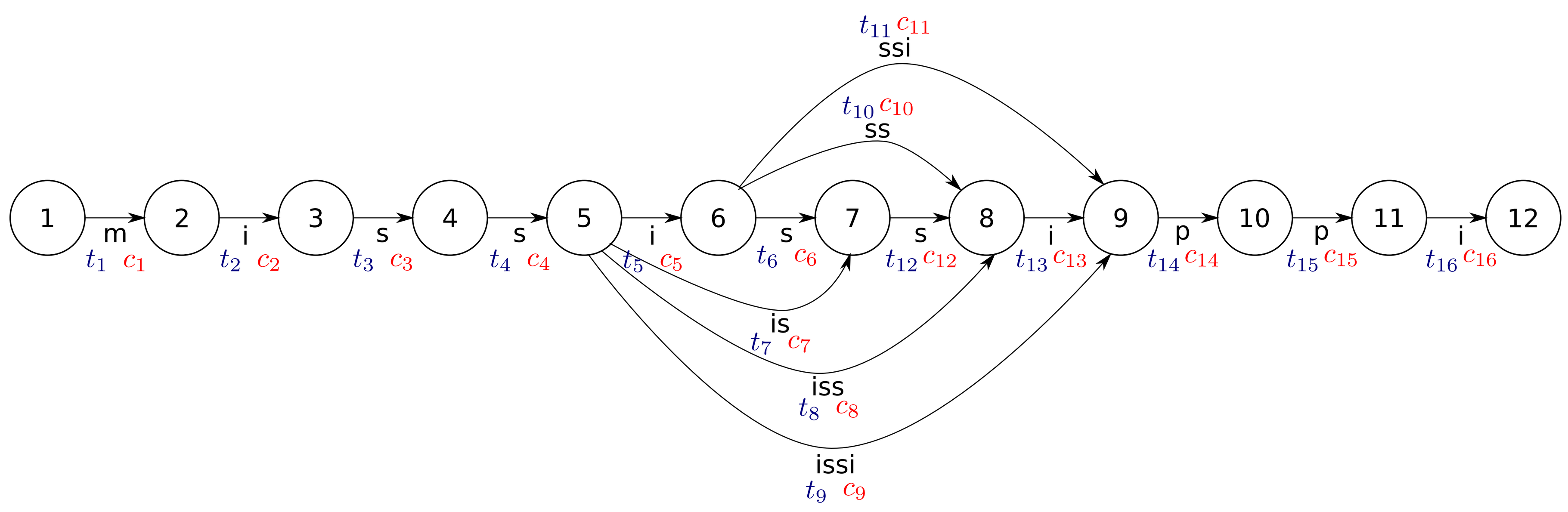
Time-Space Biobjective LZSS Parsing Strategy

The graph $\mathcal{G}(T) = (V, E)$ is defined as follows:

- $v_i \in V$ for each character $T[i]$ plus a "sink" v_{n+1};
- $(v_i, v_{j+1}) \in E$ for each substring $T[i, j]$ in the LZSS dictionary;
- a time weight and a space weight for each edge.

Let Π be the set of paths from v_1 to v_{n+1}, then:

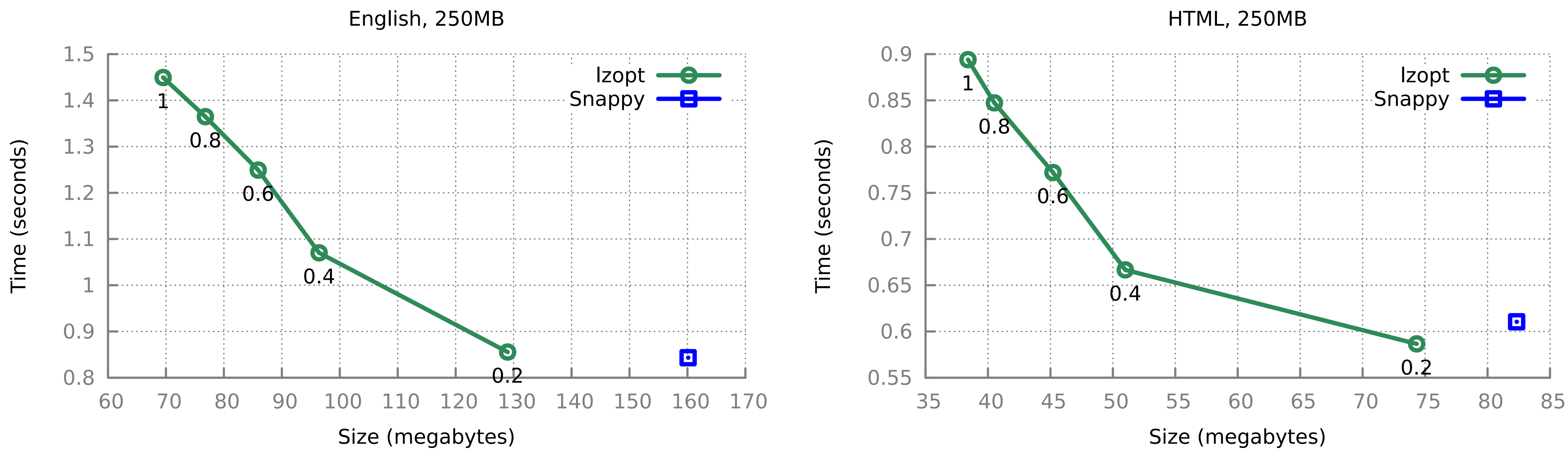
- there is a bijection between Π and LZSS parsings of T;
- the sum the edge weights on a path equals the decompression time and compression ratio of the corresponding parsing.



Time weights have been estimated by correlating decompression time of a parsing with a set of relevant parsing features like number of phrases, characters, copies which may induce cache misses, etc.

Our goal is to minimize the compressed space, given a user-specified time bound.

Since the weight-constrained shortest path problem is NP-hard in general, we employed a $O(n \log n)$ heuristic based on solving the Lagrangian Dual on the weight constraint using the cutting-plane method.



It is possible to control, in a nearly linear way, the decompression time/compression ratio trade-off. Moreover, we perform consistently better than state-of-the-art compression (like Google Snappy).