Efficient Compression of Large Route Datasets using Pattern Matching

Kishan Patel

Supervisor: Dr. Panagiotis Angeloudis

Department of Civil & Environmental Engineering, Imperial College London

INTRODUCTION

Transport users may require the fastest route between an origin and destination, which would involve the use of a shortest path algorithm, such as Dijkstra’s. Given a sizeable number of travellers seeking routes at one time-point, ‘on-demand’ path generation would become infeasible, as algorithmic runtime will create delays within a queue of route requests. This research looks into ways to retrieve information to the user ‘on-demand’ after dataset compression techniques have been applied, given that all routes within a network are generated beforehand. These algorithms can enable a reduction in processing requirements and computer storage space within embedded/mobile devices, whilst simultaneously providing useful information quickly through decompression procedures.

EXISTING COMPRESSION METHODS

A list of compression methods were initially read over and understood, to account their suitability with this research. These included the Discrete Cosine Transform (DCT), Principal Component Analysis (PCA) and CUR Matrix Decomposition. These methods were all found not to be suitable, due to their uniqueness around image/audio compression. However, the LZW Algorithm was discovered to be highly applicable, whereby the algorithm itself ‘slices’ data sequentially, and references it into a dictionary.

METHODOLOGY

RESULTS

Through the results shown, it can be seen that:

- LZW Algorithm achieves a much higher ‘extent of data compression’ compared to the SSA
- LZW Algorithm takes a much longer time to complete than the SSA

The reasons for this include the nature of the LZW Algorithm: taking each small ‘slice’ of data and checking if it has been referenced within a dictionary. Ultimately, the LZW Algorithm achieves better compression because all sliced data has a corresponding substituted character, unlike the SSA which contains a mixture of substituted characters and ‘leftover’ nodes. An important note is that processing power of the computer also has an effect upon the compression duration results. Conclusively, both algorithms achieved a substantial amount of compression for the networks analysed. The implications could mean that significant compression could allow the storage of shortest path route data within embedded/mobile devices that would not need an ‘online’ internet connection (hence function ‘offline’). The most interesting aspects of this study include how the format of data (saved within MATLAB) can drastically affect the number of bytes utilised for storage space. Information saved in ‘char’ format took up far less space than information saved within ‘cell’ or ‘numeric’ format.

ACKNOWLEDGEMENTS

I would like to thank Dr. Angeloudis and Nils Goldbeck for their help and support throughout this project.

EDC (%) = 100(1 – ALTCR) = 100 (1 – \frac{1}{CR})