# Column Partition and Permutation for Run Length Encoding in Columnar Databases

Jane Shi
University of Waterloo, j96shi@uwaterloo.ca

ACM SIGMOD SRC
June 14th , 2020

## 1. Introduction

- Effective compression is essential when in-memory columnar databases are used in Big Data applications. In particular, compression can help to load the columns faster, speed up query evaluation and reduce disk/RAM traffic.
- In this work, we focus on the Run Length Encoding (RLE). This compression algorithm is a well-known light weight compression algorithm. [1]
- We ask “how to rearrange table rows such that we can select a subset of columns to apply RLE and minimize compression size? ”
- Because the optimal solution to the row arrangement problem is NP-hard [2], we propose an incremental heuristic that identifies a subset of columns to compress, as well as an ordering of rows, in O(T log r) time[1].

1 T is the table size and r is the number of rows.

## 2. Motivation

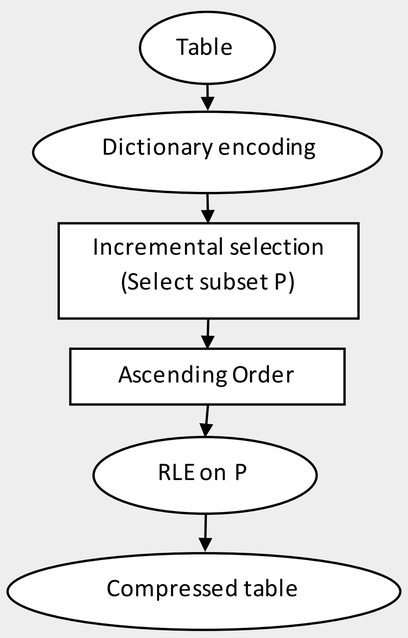
Table 1. Sample Data Table

| Country | State/Prov. | Name | Hobby |
|---|---|---|---|
| Canada | Alberta | Alice | Reading |
| Canada | Alberta | … | … |
| Canada | Alberta | Alice | Sports |
| Canada | Alberta | Bob | Studying |
| Canada | Alberta | … | … |
| Canada | British Columbia | Alice | Dancing |
| … | … | … | … |
| Canada | Saskatchewan | Zoey | Fencing |
| USA | Alabama | Alice | Fencing |
| USA | … | … | … |
| USA | Alaska | Abigail | Sports |
| … | … | … | … |
| USA | Wyoming | Zoella | Fencing |

- A new color is the start of a new run.
- In general, it is more difficult to obtain long runs as we move towards latter columns. Specifically, RLE loses its advantage on runs of length 1. Our goal is to only compress the ”desirable” columns.

## 3. Solution Overview

Figure 2. Proposed Algorithm




## 4. Problem Setup and Proposed Algorithm

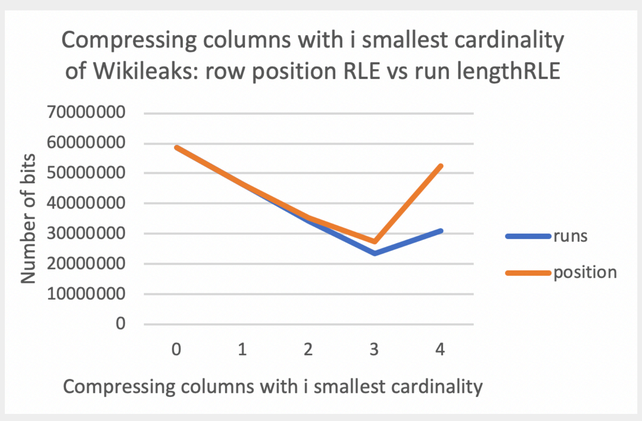
- Idea is to capture the “turning point” when columns no longer benefit from compression, so we tentatively perform compression and select desirable columns.
- We first perform dictionary encoding [3]. Then, we select P, a subset of columns (Problem 1) and determine a row order (Problem 2) that depends on these columns. Then, we reorder the rows and only compress columns in P.
- The method is inspired by [2], as the authors showed that the reordering of the columns by increasing cardinality yields the asymptotic optimal solution.
- Algorithm on Problem 1: Order columns in increasing cardinality, order rows lexicographically with respect to the column order, apply RLE to each columns, and place the columns that reduced in size into the set P.
- Algorithm on Problem 2: Leave the rest of columns unaffected, order rows lexicographically with the column order, and only compress columns in P.
- Runtime: O(T log r).

## 5. Challenges

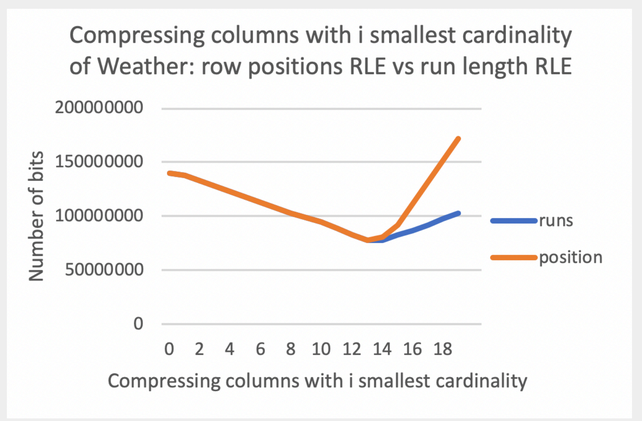
- Solution space has the size $\sum_{i=0}^{c}\binom{c}{i}i!$ where c is the number of columns.
- It is difficult to predict whether a column can benefit from compression after the row reordering.
- When lexicographically re-ordering rows with respect to different column orders, the arrangement of new columns could be vastly different.

## 6. Preliminary Experimental Results

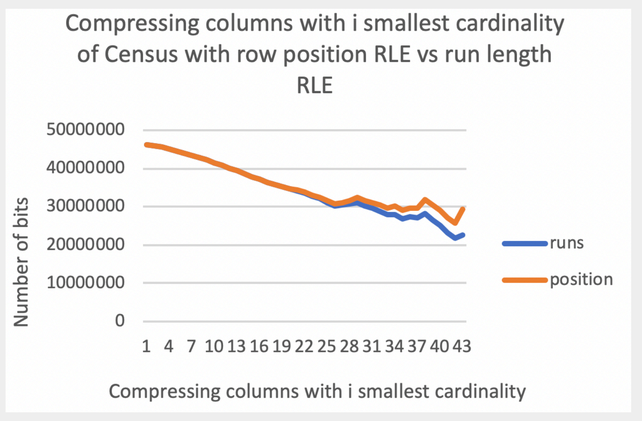
Figure 3-5. Preliminary Experimental Results on Datasets




Wikileaks dataset




Weather dataset




Census dataset

The datasets are used in [3]. We thank Dr. Daniel Lemire for providing the datasets on https://github.com/lemire/RealisticTabularDataSets.

| Dataset | No CMP | CMP all (a) | Proposed (b) | b/a(%) |
|---|---|---|---|---|
| Wikileaks | 58,469,132 | 31,073,761 | 23,398,807 | 75.30 |
| Weather | 140,120,674 | 103,067,253 | 77,566,108 | 75.26 |
| Census | 46,289,362 | 22,519,309 | 18,509,296 | 82.19 |

Table 6. Comparing Preliminary Results

## 7. References

[1] Lemke Christian, Sattler Kai-Uwe, Faerber Franz and Zeier Alexander. 2010. Speeding up queries in column stores: a case for compression DaWaK 6232, ( 2010), 117-129. https://link.springer.com/chapter/10.1007/978-3-642-15105-7_10
[2] Lemire Daniel and Kaser Owen. 2019. Reordering Columns for Smaller Indexes InformationSciences181,12(Jun.2011),2550-2570. https://dl.acm.org/citation. cfm?id=1963761
[3] Abadi Daniel,Madden Samuel and Ferreira Miguel.2006.Integratingcompression and execution in column-oriented database systems ACM SIGMOD , (Jun. 2006), 671-682. https://dl.acm.org/citation.cfm?id=1142548

ACM SIGMOD/PODS International Conference on Management of Data
June 14-19th, 2020
Portland, Oregon USA