Optimal Sort Ordering in Column Stores is NP-Complete

Abhijeet Mohapatra

December 1, 2009

In this report we study the problem of finding a lexicographic sort-ordering of a column store relation which minimizes the number of runs across attribute values. We denote this problem as *OptRLE* and show that it is NP-Complete. In section 1 we define the problem and prove its NP-hardness by reducing HAM-PATH to it. HAM-PATH is known to be NP-complete[3]. In section 2 we propose two algorithms with approximation ratios 2 and 1.5 respectively for OptRLE. An equivalent result for boolean matrices has been outlined in [4].

1 NP-completeness of OptRLE

1.1 Problem Statement

The decision problem of OptRLE is defined as follows.

Definition 1.1 *Given a relation $R(A_1, A_2, \ldots A_k)$ with n tuples (with tuple identifiers 1, 2 ... n) and RLE-constants κ_i ($1 \leq i \leq k$), the OptRLE problem asks whether there is an ordering of the tuples $r_{\pi(1)}, r_{\pi(2)}, \ldots r_{\pi(n)}$, $r_i \in R$, such that the total # of runs $\leq \kappa$. The number of runs across an attribute for an ordering is given by 1 + # of value changes in that column. The total # of runs in the relation is the sum of the runs across all attributes.*

We prove the NP-hardness of OptRLE by constructing a poly-time reduction from HAM-PATH to OptRLE.

1.2 Reduction of HAM-PATH

We first define the decision version of the HAM-PATH problem.

Definition 1.2 *Given an undirected graph$G(V, E)$, the HAM-PATH problem asks whether there exists a path that visits every vertex exactly once.*

To prove NP-completeness of OptRLE, we need to show that

1. OptRLE $\in$ NP, i.e. given a solution instance, we can verify it in polynomial time. This is true since given a relation R, n ordering π on the tuples and the RLE constant K we can verify if the total # of runs $\leq K$ or not in time linear to size of relation R.

2. OptRLE is NP-hard. In our case, we need to give a polynomial reduction of HAM-PATHto it.

Theorem 1.3 *HAM-PATH* $\leq_p$ *OptRLE*

Proof Given a problem instance of HAM-PATH, an undirected graph $G(V, E)$, construct an instance of OptRLE as follows:

1. Construct a relation R with $|V|$ tuples and $|E|$ attributes. Initially all attribute values are empty and all attributes are *unmarked*.

2. $val \leftarrow 1$.

3. $\forall (u, v) \in E$, let r_u, r_v be the corresponding tuples in R. Choose an unmarked attribute say A_i.

4. Assign $r_u = r_v = val$. Mark attribute A_i for r_u and r_v.

5. $val \leftarrow val + 1$.

6. If all edges are covered, fill the remaining attribute values with distinct values of val.

7. set RLE constant $K \leftarrow |V|.|E| - (|V| - 1)$

Note: The construction time is polynomial in size of input graph $O(|V|.|E|)$.

Claim 1.4 *The relation R with the constant ? = K is an instance of OptRLE and a polynomial reduction of HAM-PATH.*

Proof For any attribute, there are only 2 tuples that have equal values. The construction ensures this in steps 4, 5 and 6. Therefore, the maximum savings in runs per attribute is 1.

There are two possibilities:

1. (answer to OptRLE is 'YES'): Then, for some ordering π, there is a saving of 1 run per attribute and 2 runs for every tuple(except first and last tuples). Consider the path with the vertices of G taken in order of their corresponding tuple. The path always exists since for relation R there is a saving of 1 run for every attribute and 2 runs for every tuple(except first and last tuples) giving a total saving of $|V| - 1$ runs. This path visits all vertices exactly once. Therefore the answer to HAM-PATH is 'YES' too.

2. (answer to OptRLE is 'NO'): Let us assume the answer to HAM-PATH is 'YES'. Order the tuples according to the order in which the vertices appear in the Hamiltonian path. For this tuple ordering, there is a saving of 1 run per attribute and 2 runs for every tuple again(except first and last tuples). Therefore, the total savings in runs add to $|V| - 1$ and thus, the answer to OptRLE should have been 'YES', which is a contradiction. Therefore the answer to HAM-PATH is 'NO'.

Therefore, HAMPATH $\leq_p$ OptRLE. ∎

2 An Approximate solution

In the following section, we provide two algorithms with approximation algorithms (ratios 1.5 and 2 respectively) for OptRLE.

We first reduce OptRLE to an instance of the Traveling Salesman Problem.

Definition 2.1 *(decision variant) Given a graph $G(V, E)$, edge weights $w_e, e \in E$ and a TSP constant K, the TSP asks whether there exists a Hamiltonian path with $\sum_{e \in Path} w_e \leq K$.*

Given an instance of OptRLE (relation R and RLE constant K), we construct an instance of TSP as follows:

1. Construct a complete graph with $|V| = \#$ of tuples.
2. For tuples $s, t \in R$, assign $w_{s,t} \leftarrow \sum_i X_{s,t}[i]$, where $X_{s,t}[i] = 0$ if $s[i] = t[i]$, otherwise 1.
3. Set TSP constant $\leftarrow K$.

This transformation is polynomial is size of input relation.

If an answer to the TSP decision problem is 'YES' then the ordering of the tuples according to the corresponding vertices in the minimum weight Hamiltonian path, will give $\sum_{e \in TSP-path} w_e$ runs. If the answer to the TSP decision problem is 'NO', there is no ordering of tuples in R that will result in K total runs. (can be proved by contradiction).

Therefore, any α-approximation algorithm for TSP will be an α-approximation algorithm for OptRLE. The weights of the TSP problem instance behave the Triangle Inequality (since the distance metric is Hamming Distance). Two algorithms based on the construction of Minimum spanning trees have been proposed that give an approximation ratio of 1.5[2] and 2[1] respectively. We describe the algorithm that achieves an approximation factor of 2.

1. Transform the OptRLE instance into a TSP instance $G(V, E)$ as described above.
2. Let $T \leftarrow MST(G)$.

3. Duplicate the edges of T to make the obtain an Eulerian graph T'.

4. Let $P \leftarrow$ Eulerian path(T')

5. Read out the vertices in order of appearance from P removing repetitions.

6. Order the tuples in order of their corresponding vertices.

References

[1] `http://en.wikipedia.org/wiki/Travelling_salesman_problem`.

[2] Christofides, N. Worst-case analysis of a new heuristic for the travelling salesman problem, 1976.

[3] Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[4] Haddadi, S., and Layouni, Z. Consecutive block minimization is 1.5-approximable. *Inf. Process. Lett.* (2008).