**Sim-Piece+**: Efficient Time Series Data Compression

Xenophon Kitsios  
xkitsios@uab.gr  
Athens University of Economics and Business  
Athens, Greece

Panagiotis Liakos  
panagiotisliakos@uab.gr  
Athens University of Economics and Business  
Athens, Greece

Katia Papakonstantinopoulou  
katia@uab.gr  
Athens University of Economics and Business  
Athens, Greece

Yannis Kotidis  
kotidis@uab.gr  
Athens University of Economics and Business  
Athens, Greece

**ABSTRACT**

Lossy compression enhances the scalability of contemporary data collection infrastructures by reducing the volume of data that needs to be transmitted and stored. Sim-Piece is a state-of-the-art lossy compressing algorithm for time series data that outperforms competing techniques, attaining compression ratios with more than twofold improvement on average over what prior lossy algorithms can offer. In this work, we introduce Sim-Piece+, an enhanced iteration of Sim-Piece that further enhances compression ratios through the utilization of advanced data encoding techniques.

**ACM Reference Format:**


**1 INTRODUCTION**

With the emergence of the Internet of Things (IoT), enormous amounts of streaming, timestamped datasets are being generated. Sensors from smart wearables, smart cities, autonomous cars, agricultural facilities etc., produce time series data which needs to be collected centrally in order to be later analyzed [1, 2, 3, 5, 6, 8]. Time series are also encountered in other domains, such as finance, e-commerce, health care, social networks and more.

Specialized time series data compression algorithms, such as Gorilla [14] and Chimp [11], are utilized to enable Time Series Databases (TSDBs) to effectively handle extensive datasets. These algorithms are designed for applications that require precise and detailed data values. On the other hand, lossy compression algorithms provide significantly greater space savings by allowing a certain level of acceptable error within a predefined bound. These algorithms are particularly suitable for applications that put emphasis on extracting meaningful patterns and insights from the data, rather than preserving precise numerical accuracy. They find utility in tasks such as seasonality detection, clustering, forecasting, and similarity search.

Sim-Piece [9] is a lossy timeseries algorithm that is based on the popular Piecewise Linear Approximation (PLA) scheme [7]. The novel algorithm takes advantage of the resemblances among the line segments generated by the PLA algorithm to create a unified representation for multiple segments. Our experimental evaluation using multiple real and synthetic datasets [9] demonstrates significant improvements in space efficiency across a wide range of maximum error values, even when the permissible error margin is minimal. In this work we present Sim-Piece+, an extension of Sim-Piece that offers even more substantial reductions in space requirements by improving the encoding of its internal representation.

**2 SIM-PIECE ALGORITHM**

PLA algorithms represent time-series measurements using a sequence of line segments, while keeping the approximation error within a predetermined acceptable threshold. Sim-Piece is a novel approach for PLA that seeks to exploit similarities among the line segments produced. Sim-Piece starts a line segment at a quantized value close to the original one. Then, it adds more points by keeping two slopes: the highest and lowest ones that fit the data within the required approximation guarantees, until a point goes beyond them. As any of the lines between the two slopes can approximate the data points of the segment, we can find groups of segments with intersecting sets of candidate lines and represent them jointly, to reduce the overall space requirements. Figure 1 shows an example
PLA approximation with two groups of mergable segments. Each group can be represented with a single pair of (starting-value, slope) values. This allows us to represent, in this example, all five PLA segments with using just two lines. Their starting timestamps are also stored, so that we can reconstruct them entirely when decompressing them. Given a set of PLA segments, Sim-Piece computes the optimal solution to this problem, coming up with the minimum possible number of groups to represent them jointly [9].

The encoding of Sim-Piece data is presented in Table 1. Each segment is represented by a line equation \( y_i(t) = a_i(t - t_i) + b_i \), where \( a_i \) denotes the slope, \( b_i \) represents the starting value, and \( t_i \) signifies the starting timestamp. These lines are grouped based on their \( b_i \) and \( a_i \) values. To ensure successful decoding, binary counters are introduced before each group, denoted as \#b and \#a. For each merged group, the list of starting timestamps of its constituent line segments is stored. Starting values \( b_i \) are quantized based on the maximum error threshold (\( error \)) used and the resulting integers are delta-encoded. This error, the first y-intercept (\( b_1 \)) and the last timestamp of the timeseries (\( t_{last} \)) are encoded too. Sim-Piece+ is an enhanced version of the original Sim-Piece which further improves the compression ratio by taking advance of a Variable-Byte encoding technique [15]. Variable-Byte encoding is a simple and effective method used for integer compression. The encoding scheme works by representing an integer value using a variable number of bytes. The basic idea is to use the most significant bit of each byte to indicate whether it is the last byte in the representation or if there are more bytes to follow. In Sim-Piece+, the utilization of Delta encoding is incorporated alongside Variable-Byte encoding to compress starting timestamps. By combining these two techniques, the compression ratio is significantly improved.

### 3 EXPERIMENTAL EVALUATION

Figure 2 shows the average compression ratio achieved by Sim-Piece and state-of-the-art PLA techniques, i.e., PMC-MR [10], Swing and Slide [4] and Mixed [12], on a real dataset extracted from [13]. The maximum error threshold used was 0.5% and 5% of each dataset range of values. Sim-piece clearly outperforms all prior PLA techniques. Sim-piece+ manages to achieve even higher compression ratios, due to the enhanced data encoding it utilizes.

### REFERENCES


---

Table 1: Data representation for encoding Sim-Piece data.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Sim-Piece</th>
<th>Sim-Piece+</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>FLOAT</td>
<td>FLOAT</td>
</tr>
<tr>
<td>( b_1 )</td>
<td>INT</td>
<td>INT</td>
</tr>
<tr>
<td>#b</td>
<td>UINT</td>
<td>VARIABLE-BYTE</td>
</tr>
<tr>
<td>for ( i = 1; i \leq #b; i++; )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( b_i )</td>
<td>BYTE (+UINT)*</td>
<td>BYTE (+UINT)*</td>
</tr>
<tr>
<td>#a_j</td>
<td>BYTE (+SHORT)</td>
<td>VARIABLE-BYTE</td>
</tr>
<tr>
<td>for ( j = 1; j \leq #a_i; j++ )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( a_{i,j} )</td>
<td>FLOAT</td>
<td>FLOAT</td>
</tr>
<tr>
<td>#t_{i,j}</td>
<td>BYTE (+SHORT)</td>
<td>VARIABLE-BYTE*</td>
</tr>
<tr>
<td>for ( k = 1; k \leq #t_{i,j}; k++ )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_{i,j,k} )</td>
<td>UINT</td>
<td>VARIABLE-BYTE</td>
</tr>
<tr>
<td>( t_{last} )</td>
<td>UINT</td>
<td>VARIABLE-BYTE</td>
</tr>
</tbody>
</table>

* Delta Encoding

Figure 2: Average compression ratio comparison of lossy PLA approaches for two error thresholds.