



UNIVERSITÄT
LEIPZIG

Florian Speer & Nils Schröder

Graph Theory Lab Results

17.01.2025

METHODOLOGY

1. Invariant Analysis

- Identify the most promising invariants (execution time vs number of clusters)

2. Isomorphism Analysis

- Find the overall fastest invariant including the isomorphism checks

3. Isomorphism Scaling Analysis

- Investigate the asymptotic scaling

INVARIANTS

- Vertex Count
- Edge Count
- Vertex Degrees
- Rank
- Edge Label Histogram
- Node Label Histogram
- Node Label with Edges Histogram
- Edge Label with Nodes Histogram
- Weisfeiler Lehman Graph Hash (1-3 Iterations)

EDGE LABEL HISTOGRAM

```
def edge_label_histogram(G: nx.Graph): # Todo type hint
    histogram = [edge_data.get('order', '') for _, _, edge_data in G.edges(data=True)]
    return tuple(sorted(histogram))
```

NODE LABEL

```
def node_label_histogram(G: nx.Graph): # Todo type hint
    histogram = []
    for node, node_data in G.nodes(data=True):
        charge = node_data.get('charge', '')
        element = node_data.get('element', '')
        histogram.append((charge, element))
    return tuple(sorted(histogram))
```

NODE LABEL WITH HISTOGRAM

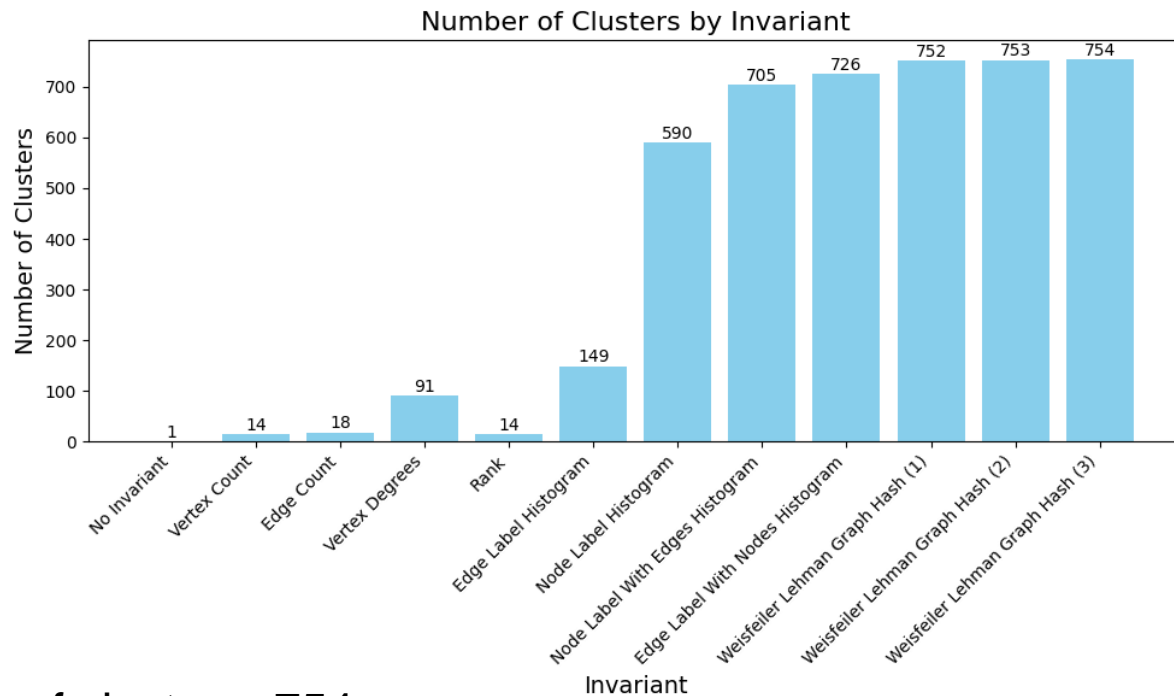
```
def node_label_with_edges_histogram(G: nx.Graph): # Todo type hint
    histogram = []
    for node, node_data in G.nodes(data=True):
        charge = node_data.get('charge', '')
        element = node_data.get('element', '')
        edge_labels = tuple(sorted(
            [edge_data.get('order', '') for _, _, edge_data in G.edges(node, data=True)]))
        histogram.append((charge, element, edge_labels))
    return tuple(sorted(histogram))
```

EDGE LABEL WITH NODES HISTOGRAM

```
def edge_label_with_nodes_histogram(G: nx.Graph): # Todo type hint
    def extract_node_label(node):
        return G.nodes[node]['charge'], G.nodes[node]['element']

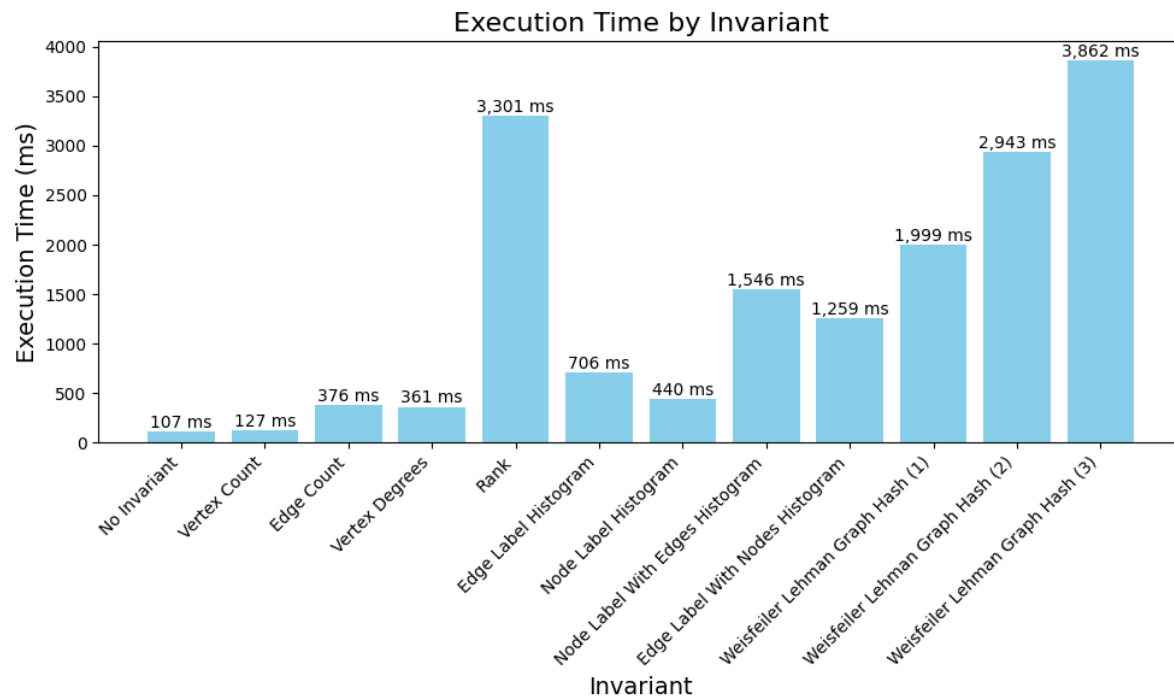
    histogram = []
    for u, v, edge_data in G.edges(data=True):
        labels = sorted([extract_node_label(u), extract_node_label(v)])
        labels.append(edge_data.get('order', ''))
        histogram.append(tuple(labels))
```

INVARIANT ANALYSIS

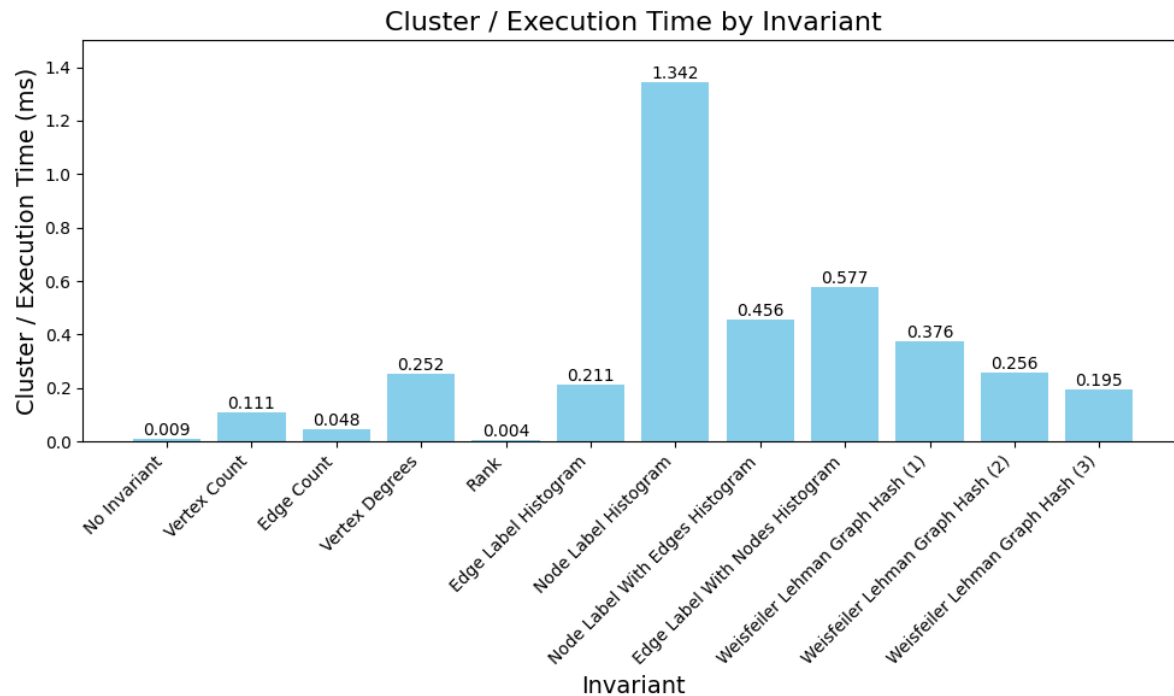


Total number of cluster = 754

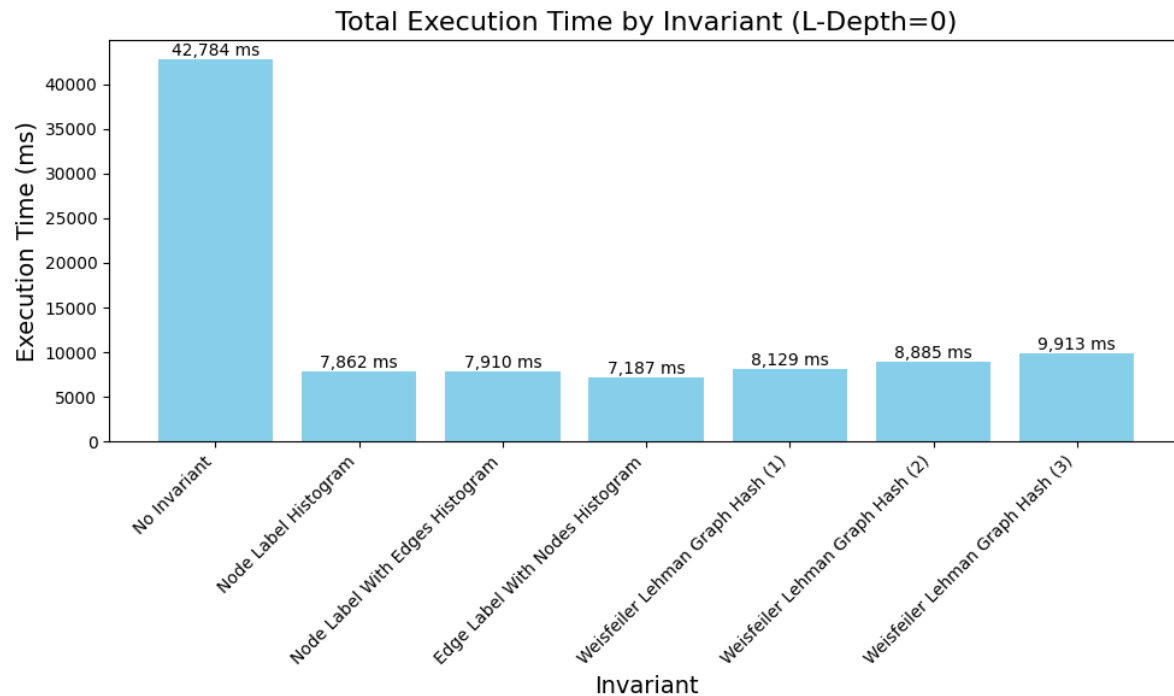
INVARIANT ANALYSIS



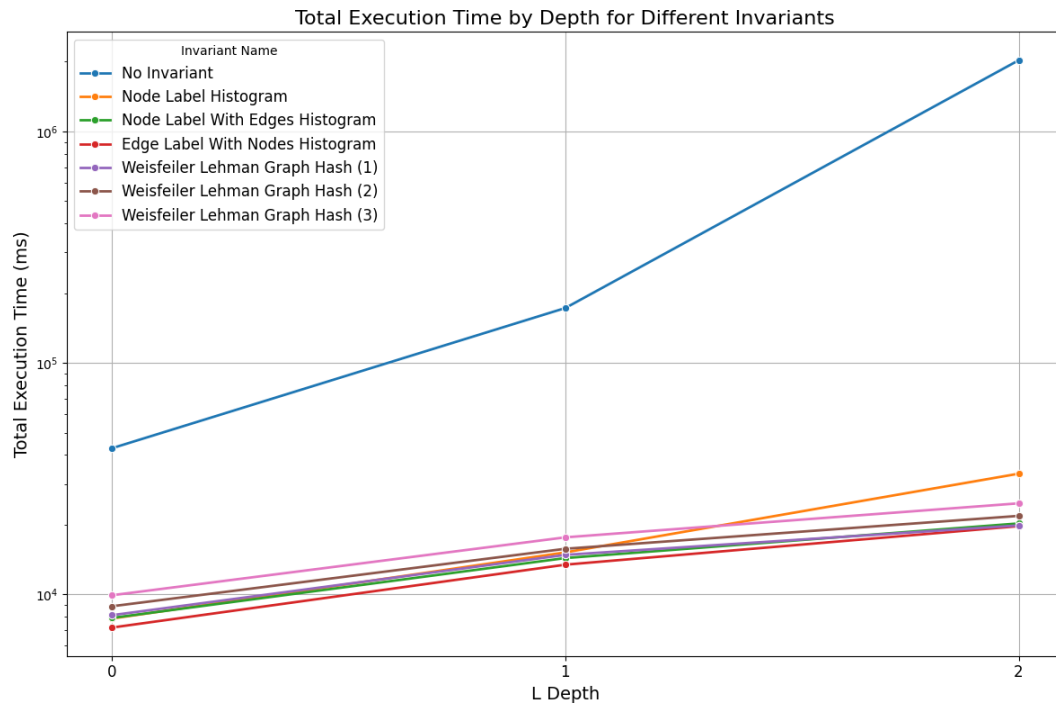
INVARIANT ANALYSIS



ISOMORPHISM ANALYSIS

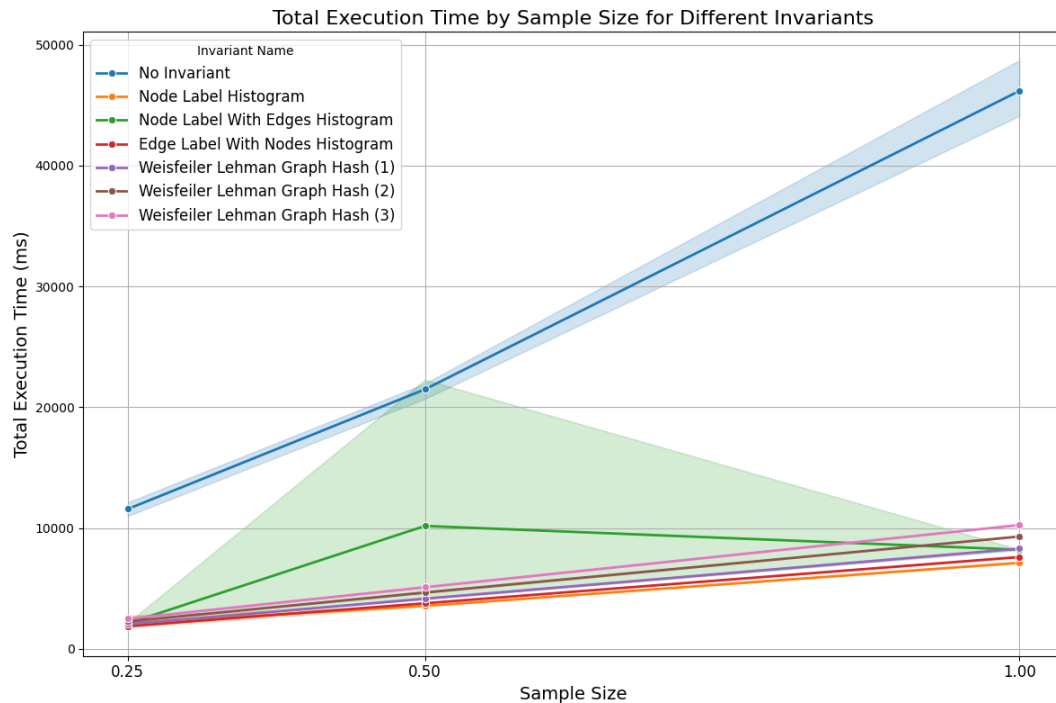


ISOMORPHISM ANALYSIS



speedup
> 16x

ISOMORPHISM SCALING ANALYSIS



L-Depth=0

CONCLUSION

- Number of clusters and number of total reactions scale differently
- Label Histograms and Weisfeiler Lehmann Graph Hash are the best invariants
 - Linear scaling for increasing number of reactions
 - Exponential scaling for increasing L Depths
- Edge Label with Nodes Histogram has the lowest total execution time
- Node Label Histogram produces many clusters in short time
- Weisfeiler Lehman Graph Hash finds with 3 Iterations all clusters for L-Depth 0-2



UNIVERSITÄT
LEIPZIG

VIELEN DANK!