```
from heapq import heappop, heappush
from profiler import profile

@profile
def hoffman_pavley(G, source, target, k, weightLabel='length'):
    k_shortest_paths = []
    paths = [(0, [source])]
    while paths and len(k_shortest_paths) < k:
        cost, path = heappop(paths)
        node = path[-1]
        if node == target:
            k_shortest_paths.append((cost, path))
        else:
            for neighbor in G.neighbors(node):
                if neighbor not in path:
                    new_cost = cost + G[node][neighbor][0][weightLabe
                    new_path = path + [neighbor]
                    heappush(paths, (new_cost, new_path))
    return k_shortest_paths
```