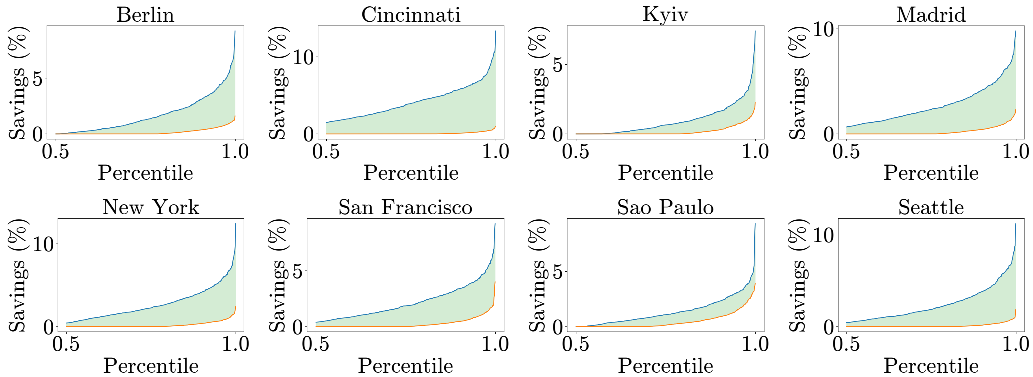

Figure 4: Distribution of travel time savings achieved by time-aware routing compared to static routing on randomly sampled instances from real-world datasets. The x-axis shows percentiles of instances, and the y-axis shows the corresponding travel time saved (in percentage). Note the long-tail distribution, indicating that significant time savings occur in a small but important subset of instances.

A Proof of Theorem 1

Theorem 1 (Hardness of TDTSP)**.** *TDTSP cannot be approximated by any $a(n)$-approximation algorithm unless P=NP, where $a(n)$ is a function that can be computed in polynomial time.*

Proof. For any Hamiltonian problem $G = (V, E)$, let $c_{ij}(t) = 1$ for $t \leq n$ and $(i, j) \in E$ in graph. Otherwise, $c_{ij}(t) = a(n) \cdot n$. The FIFO property holds. If a Hamiltonian cycle exists, the salesman can travel along the cycle with a traveling time of n. Otherwise, the salesman needs at least $a(n) \cdot n$ traveling time. So $a(n)$-approximation exists if and only if P equals NP. □

B Baseline Algorithms

This section describes the details of the baseline algorithms we use.

B.1 Greedy Algorithm

The pseudo code of the greedy algorithm is presented in Algorithm 1. The greedy algorithm chooses the earliest reachable node among unvisited nodes at each step based on the current node. The bottleneck of the algorithm is the computation speed of function c by interpolation.

Algorithm 1 Greedy Algorithm

Input: node set V, start node π_1, start time t_1, time dependent cost function c
Output: A permutation π of the node set V as a TDTSP tour.

for $i = 2, 3, \cdots, n$ **do**
  $\pi_i = \arg\min_{v \in V \setminus \pi[1:i-1]} c_{\pi_{i-1}, v}(t_{\pi_{i-1}})$
  $t_i = t_{i-1} + c_{\pi_{i-1}, \pi_i}(t_{\pi_{i-1}})$
end for

B.2 Ant Colony Optimization

The pseudo code of the Ant Colony Optimization is presented in Algorithm 2. For the visited node, we do not compute the score of them and assign value $-\infty$.

Algorithm 2 Ant Colony Optimization

Input: node set V, start node π_1, start time t_1, time dependent cost function c
Parameters: number of ants $N_{ant} = 20$, number of iterations $N_{iters} = 100$, pheromone importance $\alpha = 1$, heuristic importance $\beta = 2$, evaporation rate $\rho = 0.1$, exploitation rate $q_0 = 0.9$
Output: A permutation π of the node set V as a TDTSP tour.

for $u, v \in V$ **do**
    pheromone$[u][v] \leftarrow 1$
end for
for $i = 1, 2, \cdots, N_{iters}$ **do**
    for $j = 1, 2, \cdots, N_{ant}$ **do**
        for $k = 2, 3, \cdots n$ **do**
            for $v \in V \setminus \pi[1 : k-1]$ **do**
                heuristic$[v] \leftarrow 1/c_{\pi_{k-1} v}(t_{k-1})$
                score$[v] \leftarrow$ (pheromone$[\pi_{k-1}][v])^{\alpha} \cdot$ (heuristic$[v])^{\beta}$
                $q \sim$ Uniform$[0, 1]$
                if $q \leq q_0$ **then**
                    $\pi_k \leftarrow \arg\max_v$ score$[v]$
                else
                    $\pi_k \sim$ softmax(score)
                end if
                $t_k \leftarrow t_{k-1} + c_{\pi_{k-1}\pi_k}(t_{k-1})$
            end for
        end for
        deposit $\leftarrow 1/(t_n + c_{\pi_n \pi_1}(t_n) - t_1)$
        pheromone$[\pi_{i-1}][\pi_i] \leftarrow$ pheromone$[\pi_{i-1}][\pi_i] +$ deposit
    end for
    pheromone $\leftarrow \rho \cdot$ pheromone
end for

C Data Analysis

We show the data analysis of the remaining eight cities in Fig. 4. The cities show similar distributions to Beijing, London, Lyon, and Nairobi.