CDO: Extremely High-Throughput Road Distance Computations on City Road Networks

Shangfu Peng **Hanan Samet**

University of Maryland
{shangfu, hjs}@cs.umd.edu

Introduction

CDO: An extremely high-throughput solution, e.g., 7 million computations per second, for shortest road network distance or ETA computations using the ϵ-distance oracle on a city road network.

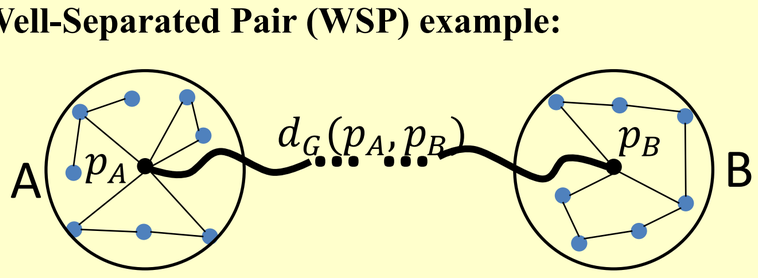
Motivation:

1) Spatial analytic queries typically perform millions of shortest distance computations in applications such as delivery, smart city analysis, local advertisement, spatial business intelligence, etc.
2) Spatial concentration property: in most applications, making use of such spatial analytic queries is typically concentrated in a small local region rather than an entire continental region.
3) Approximate shortest distance/time results are acceptable in these cases.

Distance Oracles

A Well-Separated Pair Decomposition (WSPD) on a road network consists of $\boldsymbol{O(n/\epsilon^2)}$ well-separated pairs, where n is the number of vertices, and ϵ is an error tolerance.

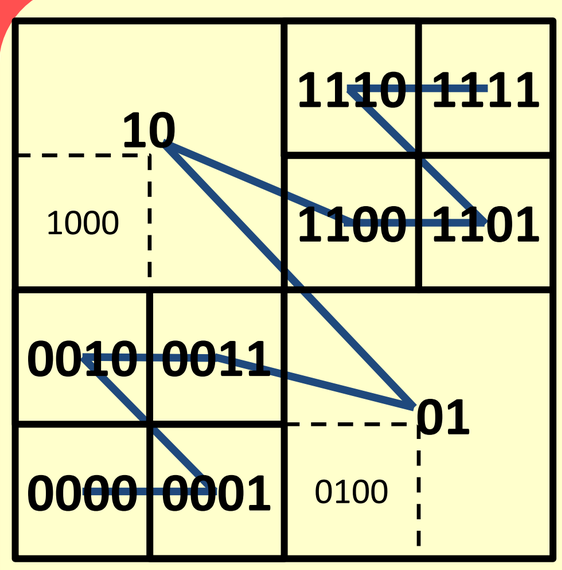
Well-Separated Pair (WSP) example:



$$\forall s \in A, t \in B, (1-\epsilon)d_G(p_A, p_B) \leq d_G(s,t) \leq (1+\epsilon)d_G(p_A, p_B)$$

Each WSP is a key-value pair where the key is the pair of two sets of vertices, A and B, satisfying the above inequalities, and the value is $d_G(p_A, p_B)$.

Storing & Querying



Key Representation:

1) Each WSP is $(A, B, d_G(p_A, p_B))$. Both A and B are PR quadtree blocks.
2) $Z_2(A)$, denoted as the Morton code of quadtree block A.
3) $Z_4(A,B)$, similar to $Z_2(Z_2(A), Z_2(B))$, but interleaving two bits at a time
4) $Z_4^0(A,B)$, padding $Z_4(A,B)$ with zeros to the same length

Example:

1) $Z_4(01,10) = 0110$ $\quad Z_4^0(01,10) = 01100000$
2) $Z_4(0000,1111) =$ $Z_4^0(0000,1111) = 00110011$

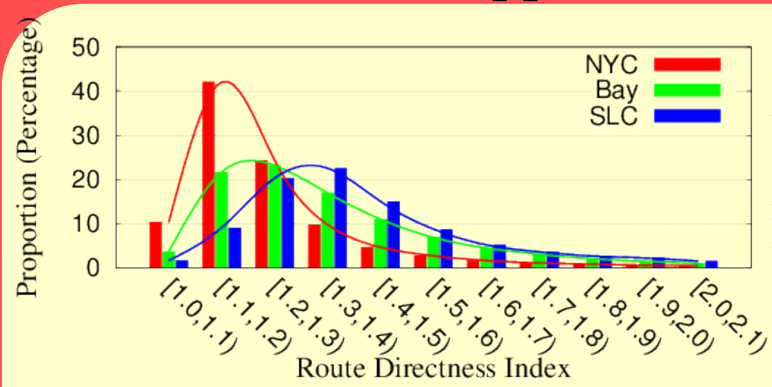
Query:

1. Load all WSPs in memory, and sort them by key, $Z_4^0(A,B)$
2. Given a source-target query (s,t), compute $Z_4^0(s,t)$
3. Binary search on all WSPs to find the WSP that $\max_{key} key \leq Z_4^0(s,t)$

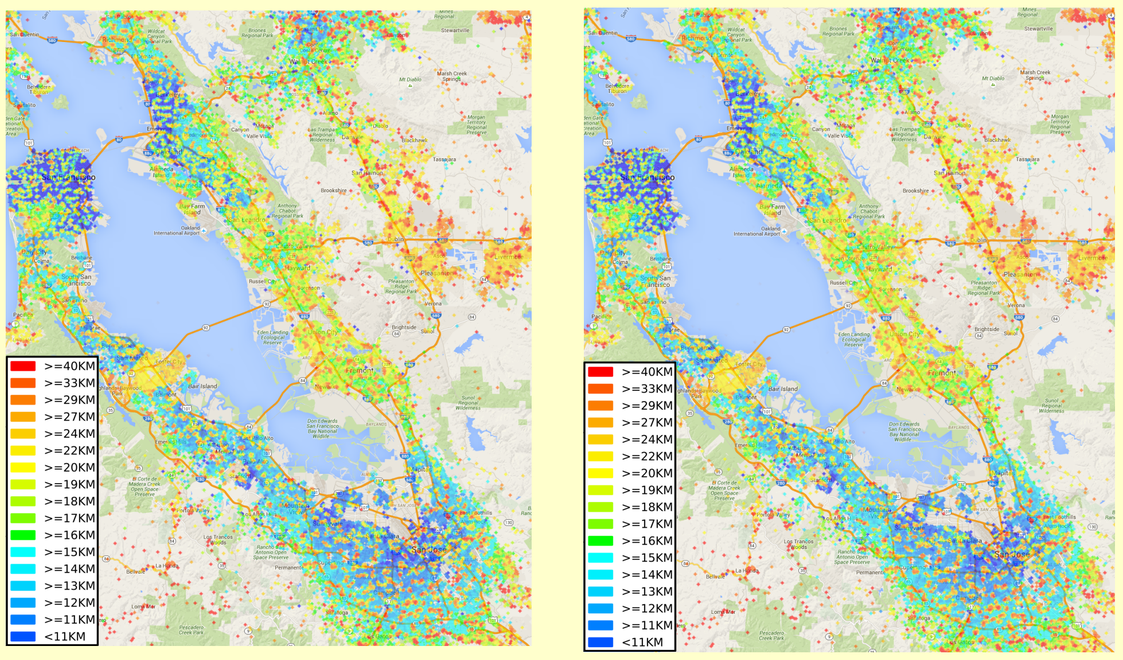
Uniqueness Property: For any source-target query (s,t), there is exactly one WSP (A,B) that contains (s,t), i.e., A contains s and B contains t.

Multi-threads improvements: only read-only operations for searching.

Application

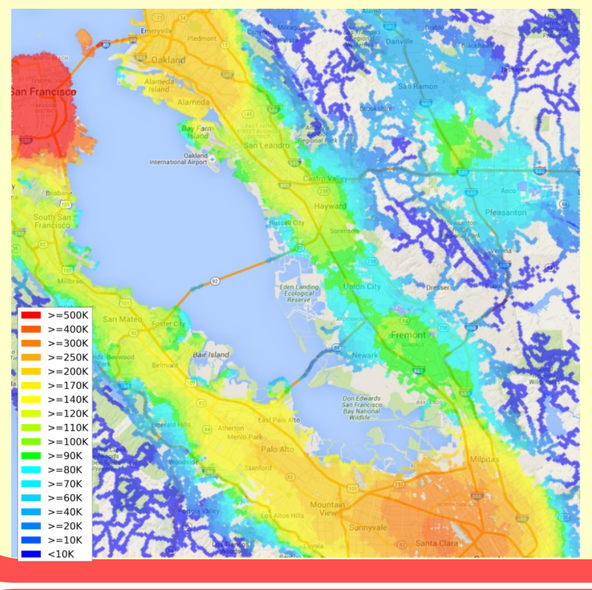


The ratio of network distance to geodesic distance. On the average, the value of this ratio for NYC is 1.213, for the Bay is 1.384, and for SLC it is 1.475.



(a) (b)

Average drive distance from home to workplace in the Bay Area region, which contains 2.1 million source-target pairs from LEHD: (a) results computed by CDO in 0.3 secs; (b) results computed by CH in 15 mins. Results in (a) are almost the same as (b). Although the distance values yielded by the distance oracles are ϵ -approximate, with $\epsilon = 0.1$, they are definitely sufficient for such analytic queries.



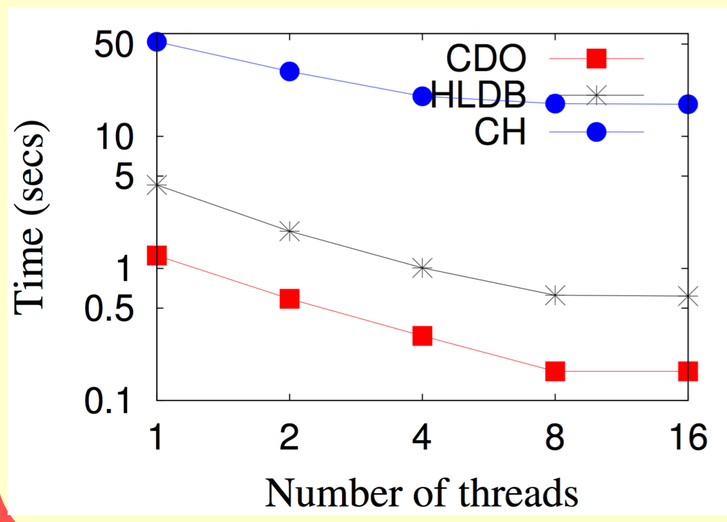
Nearby job opportunities (e.g., within 10 kms) for each census block in the Bay Area, requiring 120 million distance computations, where CDO finished it within 18 seconds and CH needs more than 10 hours.

Performance

Comparisons: HLDB in memory and Contraction hierarchies (CH)

Datasets: Bay Area road network with 781K vertices and New York City road network with 407K vertices from OpenStreetMap.

CDO Size:
1) The number of WSPs for the Bay Area road network is 392M with ϵ =**0.1** , which uses 4.3GB of memory.
2) The number of WSPs for the New York City road network is 629M with ϵ =**0.05**, which uses 7.1GB of memory



Showing the time consumption for one million random source-target pairs on the Bay Area road network, varying with number of threads.

CDO achieves 0.16 seconds with 8 threads for one million distance computations.