Hierarchical Cut Labelling - Scaling Up Distance Queries on Road Networks

Muhammad Farhan[1], Henning Koehler[2], Robert Ohms[1], Qing Wang[1]

[1]Graph Research Lab, School of Computing, Australian National University, Australia

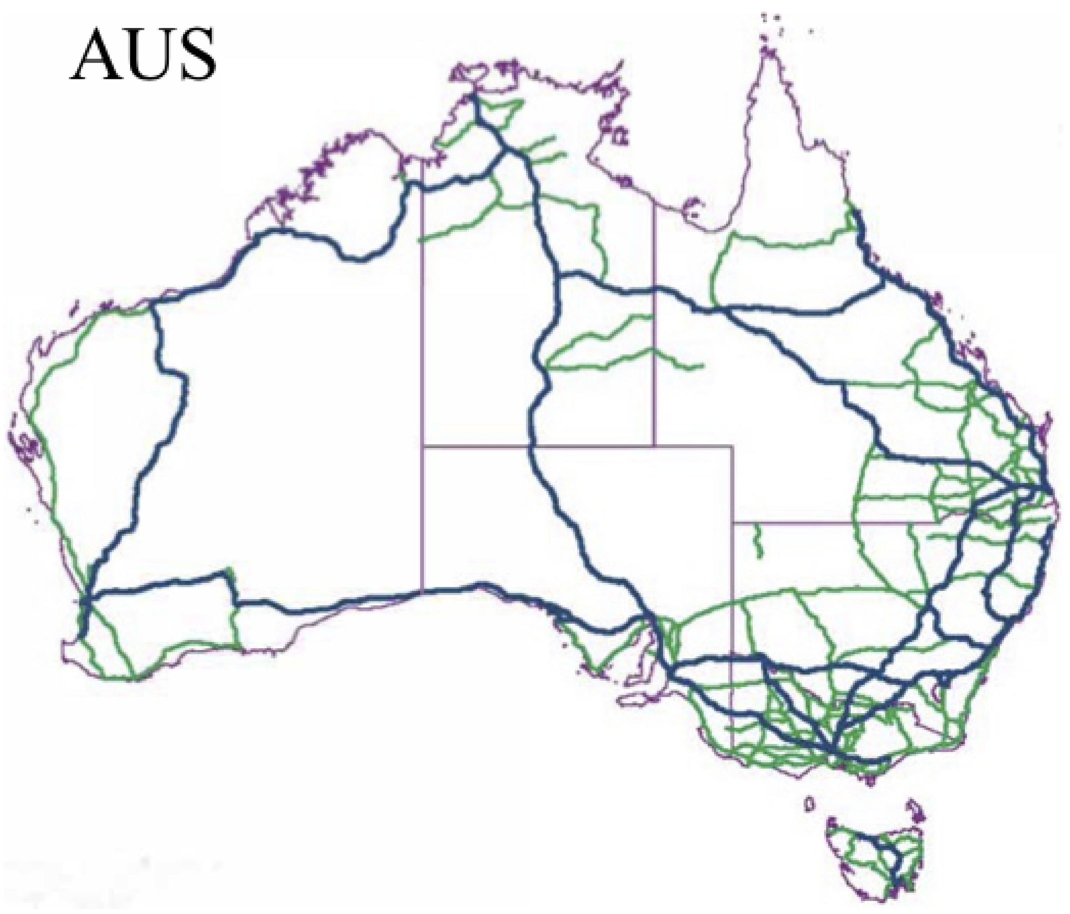
[2]School of Mathematical and Computational Sciences, Massey University, New Zealand

1. Overview

HC2L is a scalable algorithm for answering distance queries on large road networks. Our code is publicly available on **GitHub**.

HC2L-Paper HC2L-Code

2. Road Networks



Let $G = (V, E, \omega)$ be a weighted graph, where

- V represents intersection,
- E represents roads between intersections,
- ω is a weight function.

How to efficiently compute the weight of a shortest path between vertices on road networks?

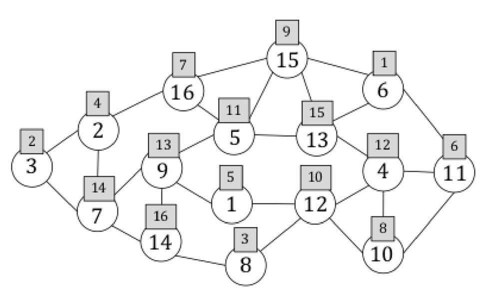
3. 2-Hop Labelling Approaches

A distance labelling L over G is a *2-hop labeling* if for any two vertices $s, t \in V$,

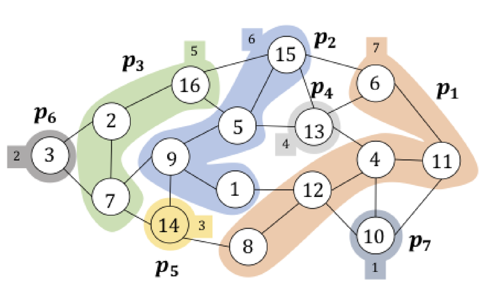
$$d_G(s,t) = \min_{u \in L(s) \cap L(t)} \{d_G(s,u) + d_G(u,t)\}.$$

Three popular techniques that exploit hierarchical structures to compute 2-hop labellings:

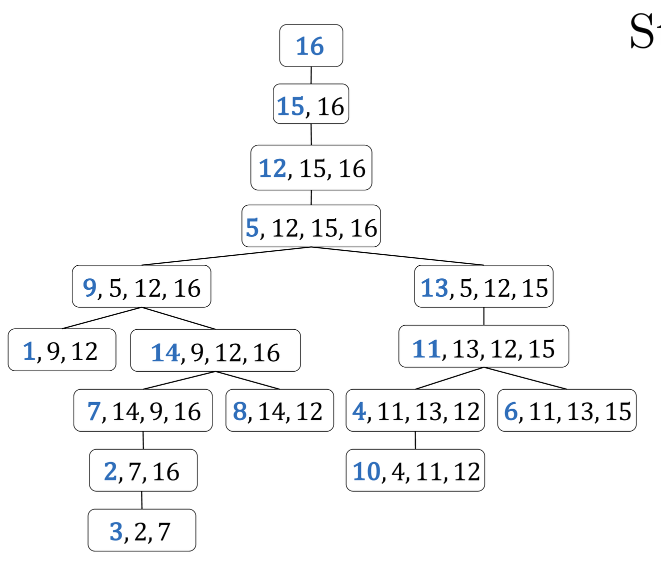
1. Hub-based Labellings;
2. Highway-based Labellings;
3. Tree-Decomposition Labellings.



Vertex Ordering



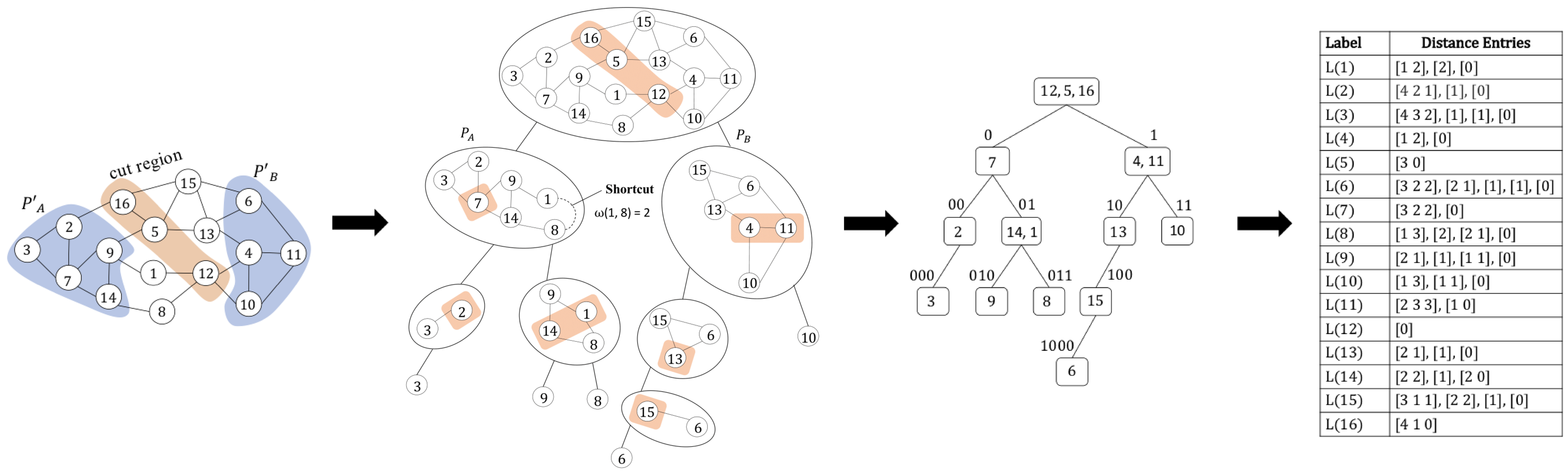
Highway Structure



Tree Decomposition

4. High-Level Framework - An Illustration

There are three main components: *hierarchical balanced cuts*, *balanced tree hierarchy*, and *hierarchical cut 2-hop labelling* (HC2L).



5. Main Ideas in Our Solution

Hierarchical balanced cuts. Our algorithm recursively bisects a graph G in two steps:

1. *Balanced partitioning*: partition $G' \subseteq G$ into two partitions connected via a *cut region*;
2. *Minimal vertex cuts*: find a minimal vertex cut within the cut region.

Balanced tree hierarchy. A binary tree $H_G = (\mathcal{N}, \mathcal{E}, \ell)$ with tree nodes $\mathcal{N}$, tree edges $\mathcal{E}$, and a total surjective function $\ell : V(G) \to \mathcal{N}$, satisfying the following two conditions:

1. For any internal tree node $N_i \in \mathcal{N}$, its left and right subtrees are balanced:

$$|\textsc{Left}(N_i)|, |\textsc{Right}(N_i)| \leq (1 - \beta) \cdot |\textsc{Subtree}(N_i)|.$$

2. For any two vertices $s, t \in V$, their lowest common ancestor (LCA) in H_G contains at least one vertex on a shortest-path between s and t.

Hierarchical cut 2-hop labelling (HC2L). Let $\preceq$ be the *vertex quasi-order* defined by a *balanced tree hierarchy* on $V(G)$. A distance labelling L_G over G is a *hierarchical cut 2-hop labelling* (HC2L) w.r.t. H_G if it satisfies the following conditions:

1. For any label $L(v)$, $v \preceq u$ holds for any vertex $v \in L(u)$;
2. For any two vertices $s, t \in V$, there exists $r \in LCA(s,t)$ such that $(r, \delta_{sr}) \in L(s)$, $(r, \delta_{tr}) \in L(t)$ and $\delta_{sr} + \delta_{tr} = d_G(s,t)$.

Note. For any two vertices $s, t \in V$, $LCA(s,t)$ can be computed as the number of leading zeros of the XOR of the bitstrings of s and t.

6. Results of Query Time, Labelling Size, and Construction Time

Network	Size		Query Time [μs]				Labelling Size				Construction Time [s]				
	$\|V\|$	$\|E\|$	HC2L	H2H	PHL	HL	HC2L	H2H	PHL	HL	HC2L	HC2L^p	H2H	PHL	HL
NY	0.3M	0.7M	**0.225**	0.432	0.983	0.765	**144 MB**	341 MB	320 MB	233 MB	15	**6**	16	34	32
BAY	0.3M	0.8M	**0.220**	0.563	0.707	0.665	**113 MB**	339 MB	235 MB	219 MB	12	**4**	12	18	27
COL	0.4M	1M	**0.351**	0.750	0.909	0.720	236 MB	**217 MB**	403 MB	341 MB	27	**12**	21	38	45
FLA	1M	3M	**0.371**	0.754	0.965	0.827	**487 MB**	1.25 GB	1.14 GB	907 MB	68	**23**	46	121	137
CAL	2M	5M	**0.442**	1.125	1.106	0.958	**1.24 GB**	3.87 GB	2.58 GB	1.78 GB	215	**57**	146	327	318
E	4M	9M	**0.555**	1.241	1.671	1.218	**3.37 GB**	9.81 GB	8.44 GB	4.74 GB	654	**163**	409	1,578	1,149
W	6M	15M	**0.583**	1.382	1.661	1.163	**5.71 GB**	18.3 GB	13.5 GB	7.50 GB	1,197	**261**	702	2,314	1,654
CTR	14M	34M	**0.760**	1.630	2.503	1.613	**24.4 GB**	73.9 GB	55.9 GB	25.5 GB	6,203	**1,658**	4,029	15,882	7,591
USA	24M	58M	**0.737**	1.940	2.389	1.663	45.1 GB	155 GB	95.6 GB	**44.7 GB**	11,203	**1,977**	7,737	26,515	13,157
EUR	18M	43M	**0.922**	2.414	2.239	1.673	44.1 GB	160 GB	70.9 GB	**34.1 GB**	12,242	**3,083**	9,194	20,466	8,728

7. Results of Query Performance (Varying Distances)