Construction of Independent Spanning Trees in Dual Cube

Dual Cube 中獨立生成樹之建構

指導教授：謝孫源 教授
專題成員：劉睿珮

1. Research Motivation and Objectives

Independent Spanning Trees (ISTs) are important graph structures for improving the reliability of interconnection networks and fault-tolerant communication by providing multiple independent paths from each node to the same root.

This project focuses on constructing node-ISTs in Dual Cube, a Hypercube-based network topology with fewer required links.

2. Background

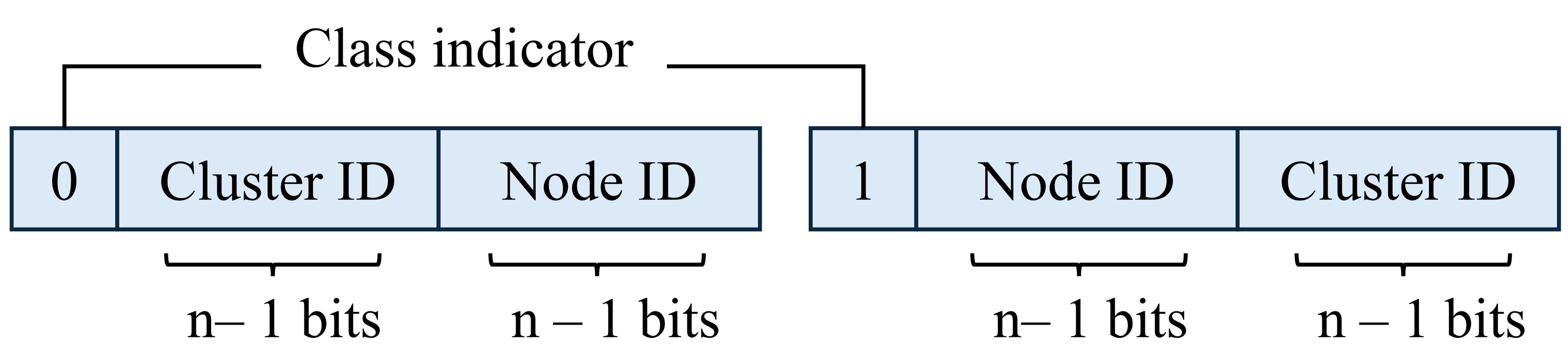
2.1 Node-Independent Spanning Trees (ISTs)

A set of node-ISTs in a graph G satisfies:

- They are all rooted at the same root r.
- For any $v \neq r$, the paths from node v to r in different trees are internally node-disjoint.

2.2 Dual Cube

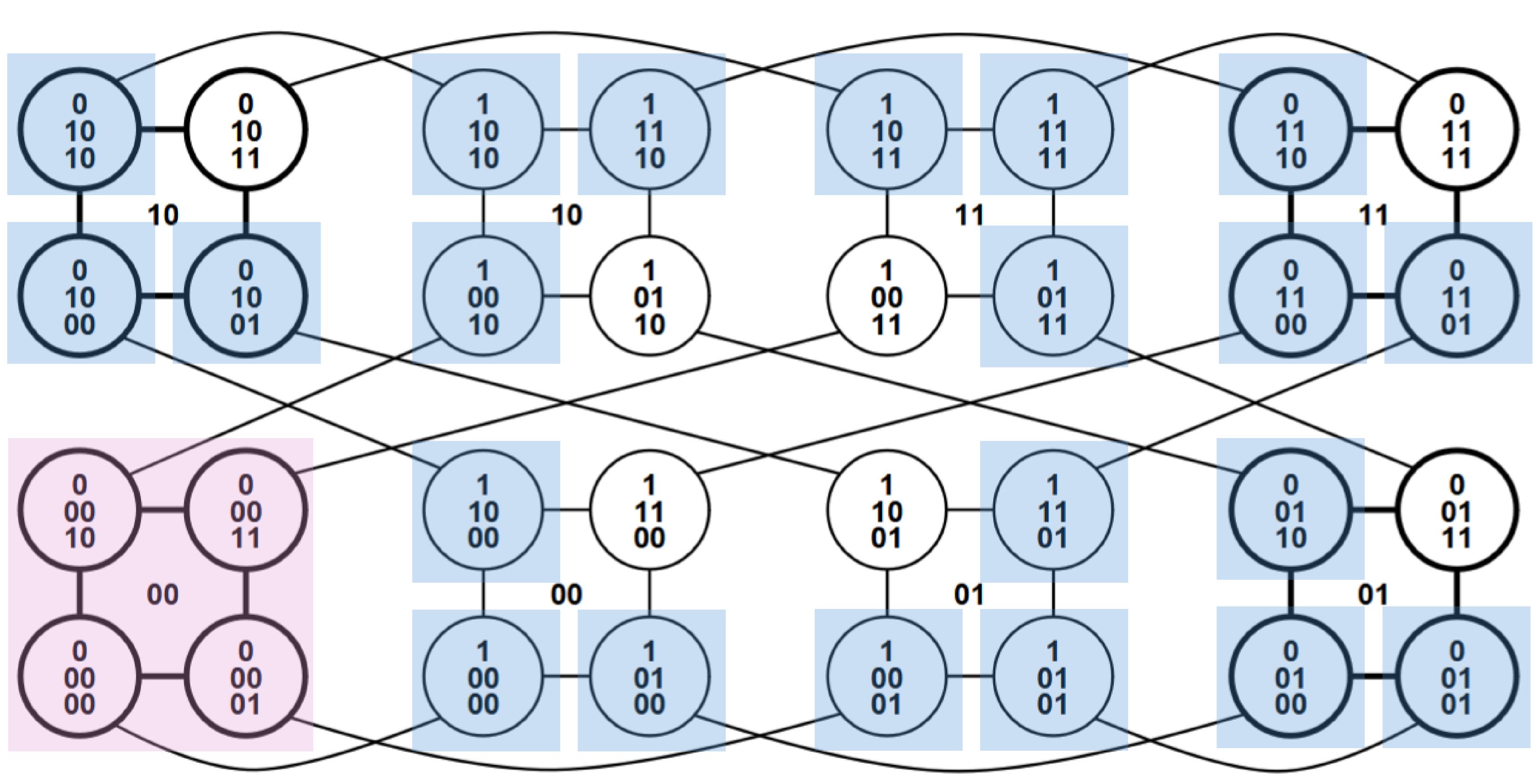
Dual Cube is a variant of Hypercube. An n-connected Dual Cube (DQ_n) is an undirected graph on the node set $\{0, 1\}^{2n-1}$.



There is an edge between two nodes if:

- The two nodes differ only in the class indicator. The edge is called cross edge.
- The two nodes are in the same class and cluster, and differ in exactly one bit of node ID.

The nodes are partitioned into two classes, each consisting of multiple clusters. Each cluster is isomorphic to Q_{n-1}.



3. Proposed Algorithm

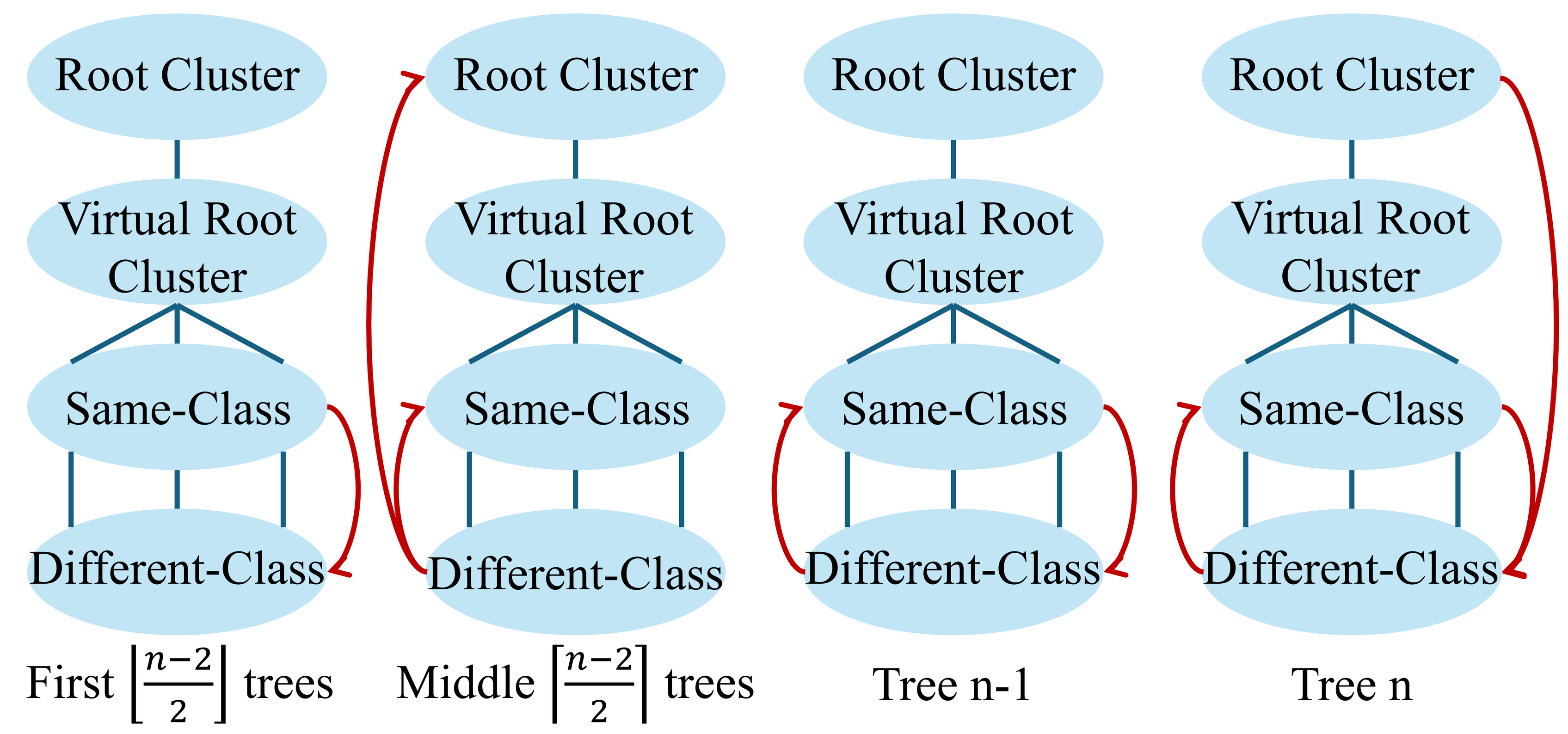
The proposed algorithm constructs n trees in DQ_n by extending existing Hypercube IST construction method to the cluster-based structure of DQ_n.

3.1 Node Classification

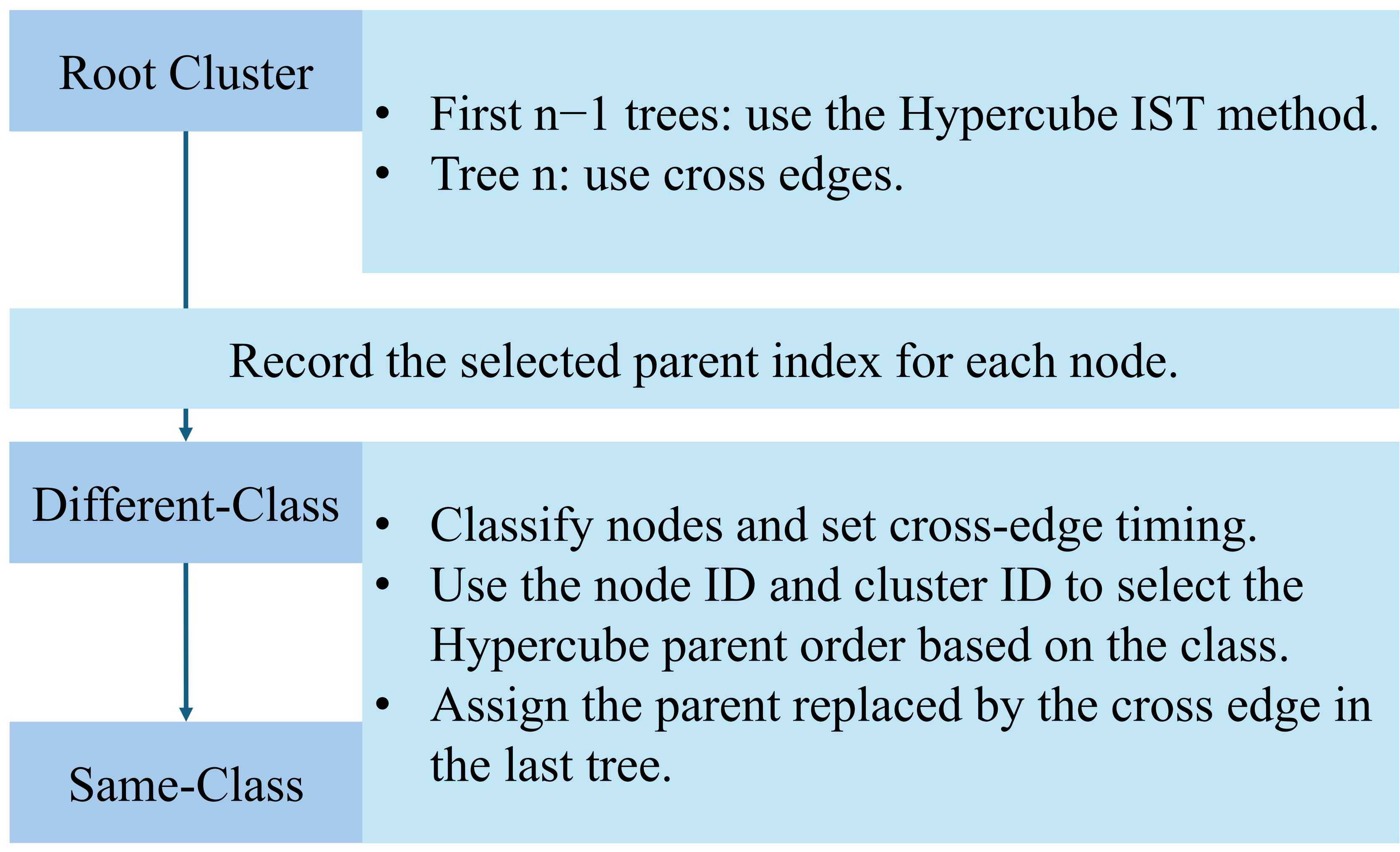
Nodes are classified by bit differences between their node ID and cluster ID to determine when cross edges are used.

- Local root: the representative node that other nodes in the same cluster move toward.
- Non-root node: not a local root in any tree.

3.2 Cluster-Level Framework



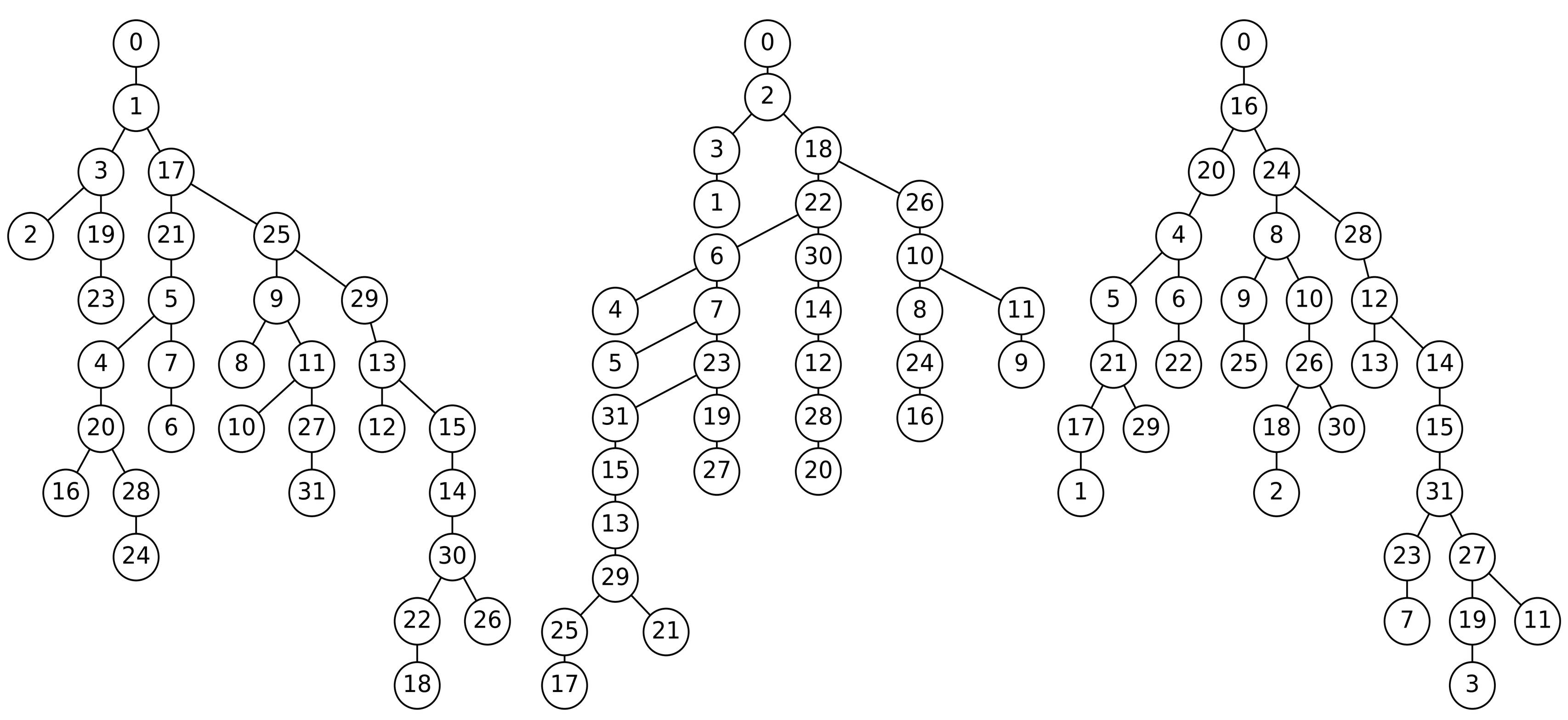
3.3 Overall Construction Algorithm



4. Results and Conclusion

The proposed algorithm was implemented using Python.

4.1 DQ_3 Construction Result



4.2 Conclusion

The algorithm assigns a parent for each node in each tree. Hence, the time complexity is $O(n|V|) = O(n \cdot 2^{2n-1})$.

The proposed algorithm can construct the first n−1 trees that satisfy the IST property. The construction rule for the final tree is still under refinement.

5. Future Work

Future work will focus on refining the construction rule for the final tree, improving cross-edge timing and parent-order selection, and establishing a formal proof of correctness for the proposed algorithm.