

Siminet Steps

Let Gcmp refer to the comparison graph (graph we are comparing against the reference) and Gref refer to the reference graph.

1. *Substitution.* Find the nodes shared between the graphs, and shift in nodes of Gcmp that are close to the node of Gref. Nodes are considered to be shared if they are approximately equivalent to each other (correspond to roughly the same location). If we can't find an approximately equivalent node in the Gref, then we shift in the closest node in Gcmp (within a certain radius). Mark nodes that were considered to be equivalent or were shifted inwards.

Substitution cost = Euclidean distance between Gcmp node and Gref node.

2. *Insertion.* If we can't find any neighboring node in Gcmp for a node in Gref, then we add a new node.

Insertion cost = Radius around which we would search for nodes to shift in

This cost is chosen so the cost of insertion is greater than that for substitution.

3. *Deletion.* Delete any nodes that were not considered to be equivalent or shifted inwards.

Deletion cost = Insertion cost

4. *Edge distance.* Find the absolute difference between the weights of every edge in Gref and its corresponding edge in Gcmp. The corresponding edge in Gcmp consists of nodes that were approximately equivalent – if we can't find a corresponding edge, then the cost should be the absolute value of the weight of Gref's edge.

Implementation Ideas

First Version

1. "Sweep" around every node in Gref with radius R, and look for the closest node in Gcmp. If the closest node falls within some equivalence radius (which is to be small), then we mark that node as equivalent – otherwise, we mark the node as shifted. We update an equivalency mapping structure (a dictionary) and a set of shifted or equivalent nodes. Based on whether an equivalency was established or a node was shifted, we update the cost. If no nodes are found, then we perform an insertion and add that to the cost.
2. Delete nodes that are not in the shifted or equivalent nodes set.
3. Compute the edge distance using the equivalence mapping to find corresponding edges in Gcmp, if they exist, and update the weights correspondingly.

Second Version

- For every node in Gcmp, search for partners in Gref
 - If we have a close partner within *equivalency radius* then we have equivalence
 - If we have a partner within the *search radius*, then we have substitution
 - No partner means deletion
- Number of nodes to insert = Number of Gref nodes - (Number of equivalence + Number of substituted nodes)
- Use equivalency mapping to find edge weights