Algorithms

Algorithm 1. Hierarchical arc-merging (HAM) Algorithm.

Input: network G.
Output: network list L.

$NetLevel \leftarrow 0$
$L \leftarrow [\,]$
For v_i in V:
    create neighbor set $\Gamma(i)$ of v_i
    calculate node degree k_i of v_i
For $e_{ij} = (v_i, v_j)$ in E:
    calculate similarity w_{ij} of e_{ij}
append G to network list L: $L \leftarrow L + [G]$
$NetLevel \leftarrow NetLevel + 1$
While true:
    create an empty super-node network H
    If $NetLevel == 1$:
        /* Original HAM network phase */
        create H initial member-node information
        classify E into three edge classes: E^W, E^B and E^S
        If E^W is empty:
            break
        process weighted-edge E^W to construct H using S1-1
        process bridge-edge E^B to construct H using S1-2
        process sink-edge E^S to construct H using S1-3
    Else:
        /* Super-node HAM network phase */
        create previous level network G member node information
        classify E into two edge classes: $E^{\Delta Q}$ and E^P
        If $E^{\Delta Q}$ is empty:
            break
        process deltaQ-edge $E^{\Delta Q}$ to construct H using S1-4
        process passed-edge E^P to construct H using S1-5
    refine H member node information
    calculate H modularity using member-node information
    If $NetLevel > 1$:
        $\Delta Q = Q^H - Q^G$
        If ΔQ is less than threshold:
            break
    copy H as G: $H \leftarrow G$
    append G to network list L: $L \leftarrow L + [G]$
    $NetLevel \leftarrow NetLevel + 1$
Return L