

Recursive Mincut algorithm.

Recursive Mincut (G , threshold, isFirst)

$M = \{\}$ // An empty set of graphs

$N = |V(G)|$ // number of vertices in G

if ($n < 2$)

 return $M = M \cup \{G\}$ // not enough vertices to subdivide

endif

$c = \text{Mincut}(G)$ // call Minimum capacity cut algorithm

$t = \text{sum}(c)$ // find the value of the cut

$t = t/|E(c)|$ // normalise the cut value by the number of edges in the cut list

if ($(t \leq \text{threshold})$ or (isFirst))

$l = \text{createGraph}(G, c, \text{"LEFT"})$ // create graph from vertices and edges in G to left of cut c

$r = \text{createGraph}(G, c, \text{"RIGHT"})$ // create graph from vertices and edges in G to right of cut c

$M = M \cup \{\text{RecurseMincut}(l, \text{threshold}, \text{FALSE})\}$

$M = M \cup \{\text{RecurseMincut}(r, \text{threshold}, \text{FALSE})\}$

else

$M = M \cup \{G\}$

endif

return M