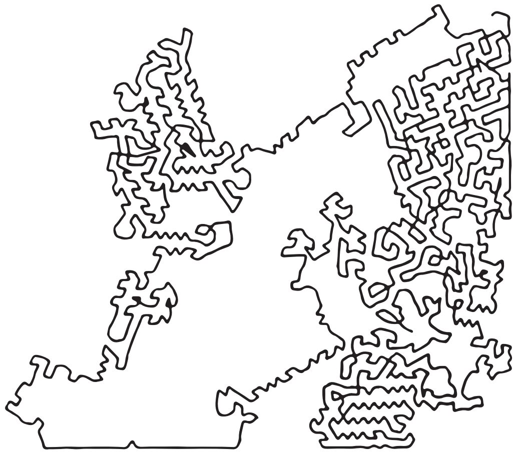


INFORMED MILLING

SCRIPT BOOKLET

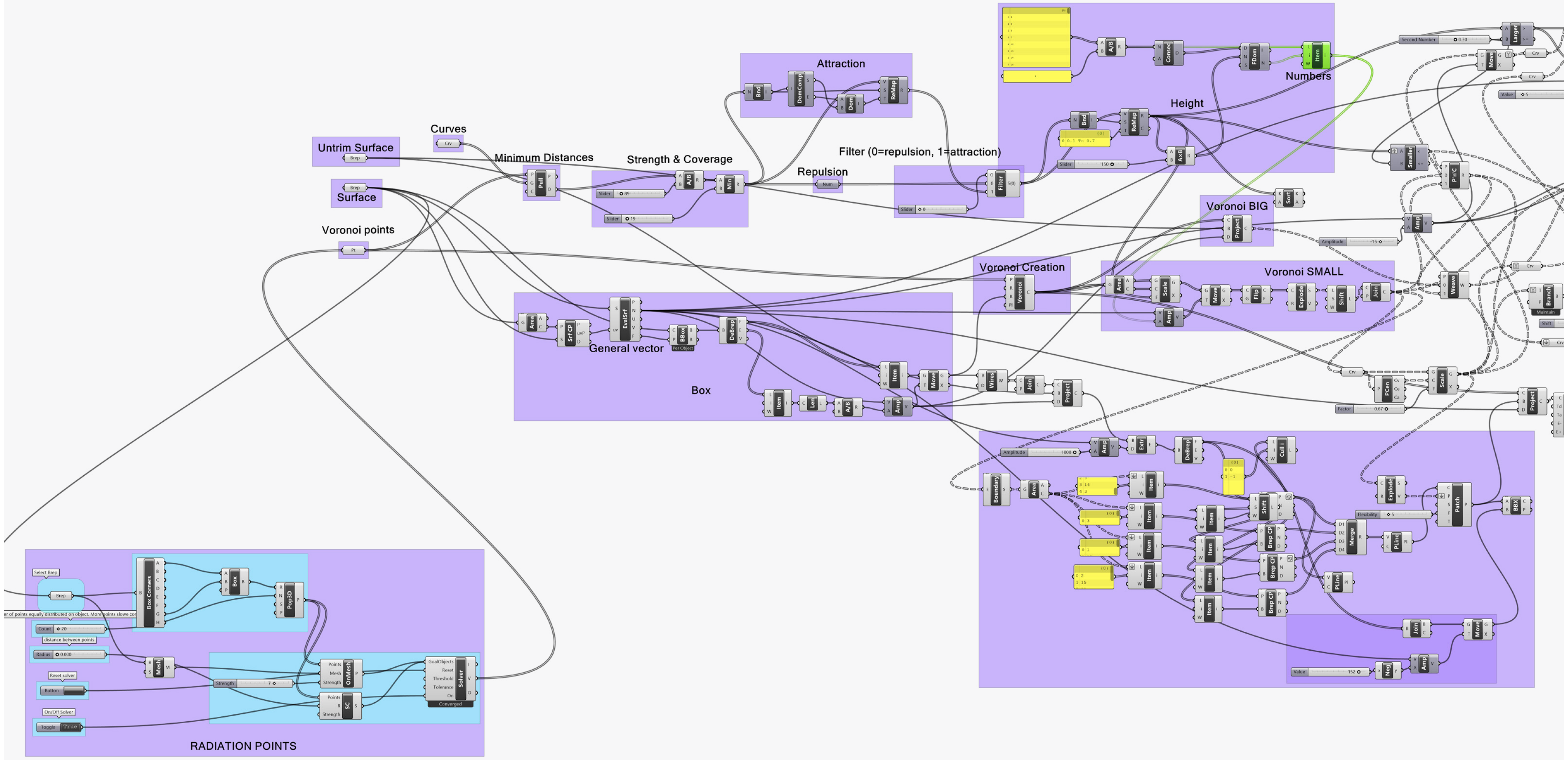
MSc 2 . GROUP 1 . Hovav, Pasveer, Leemeijer, Kisa, Sidiropoulo, Buno Heslinga

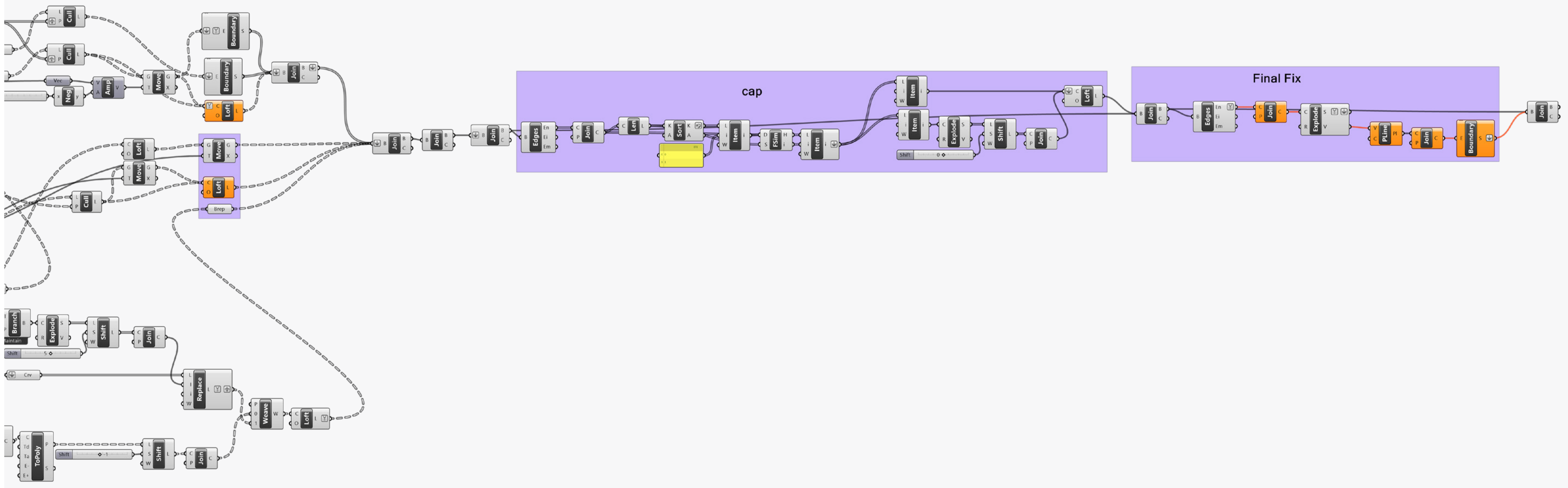
Contents

“the craft of building with **timber** is lost,
but it could be reinvented through **robotics**”

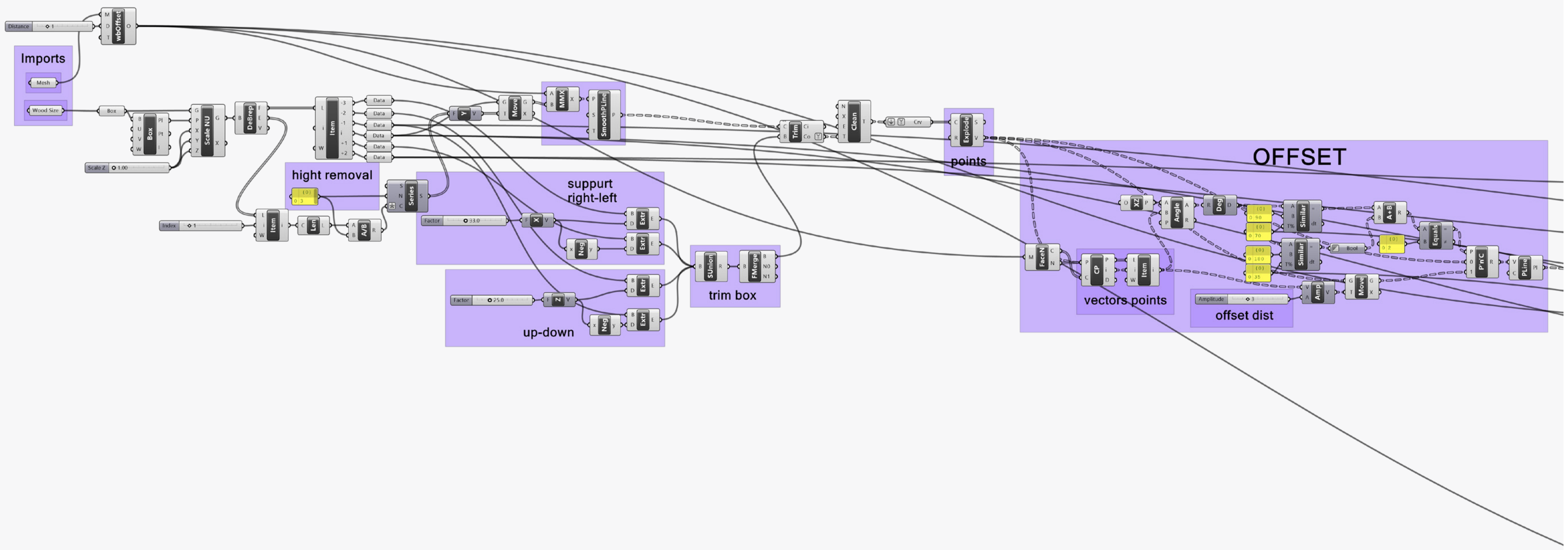
- Jan Dierck. Foster+Partners

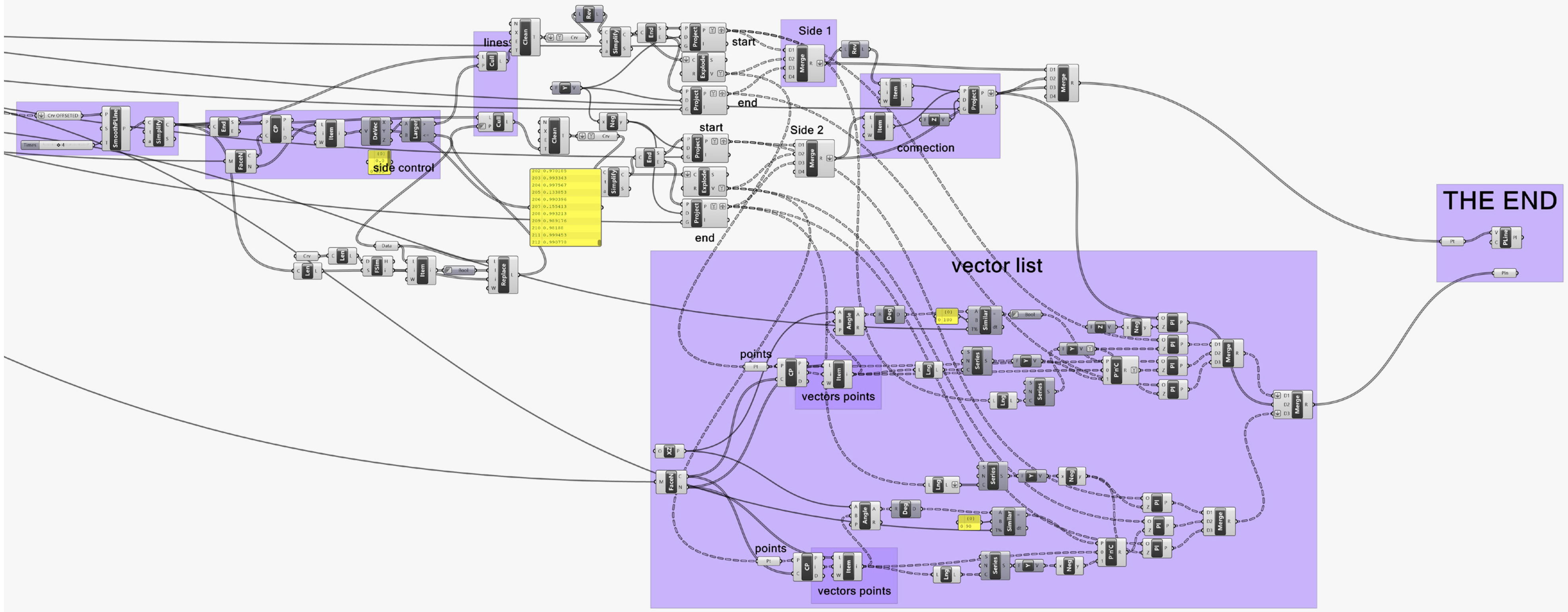
shape optimization




cap
Final Fix
Cull
Boundary
Join
Move
Loft
Edges
Sort
Item
Explode
Shift
PLine
Weave
Replace
TopPoly
Branch

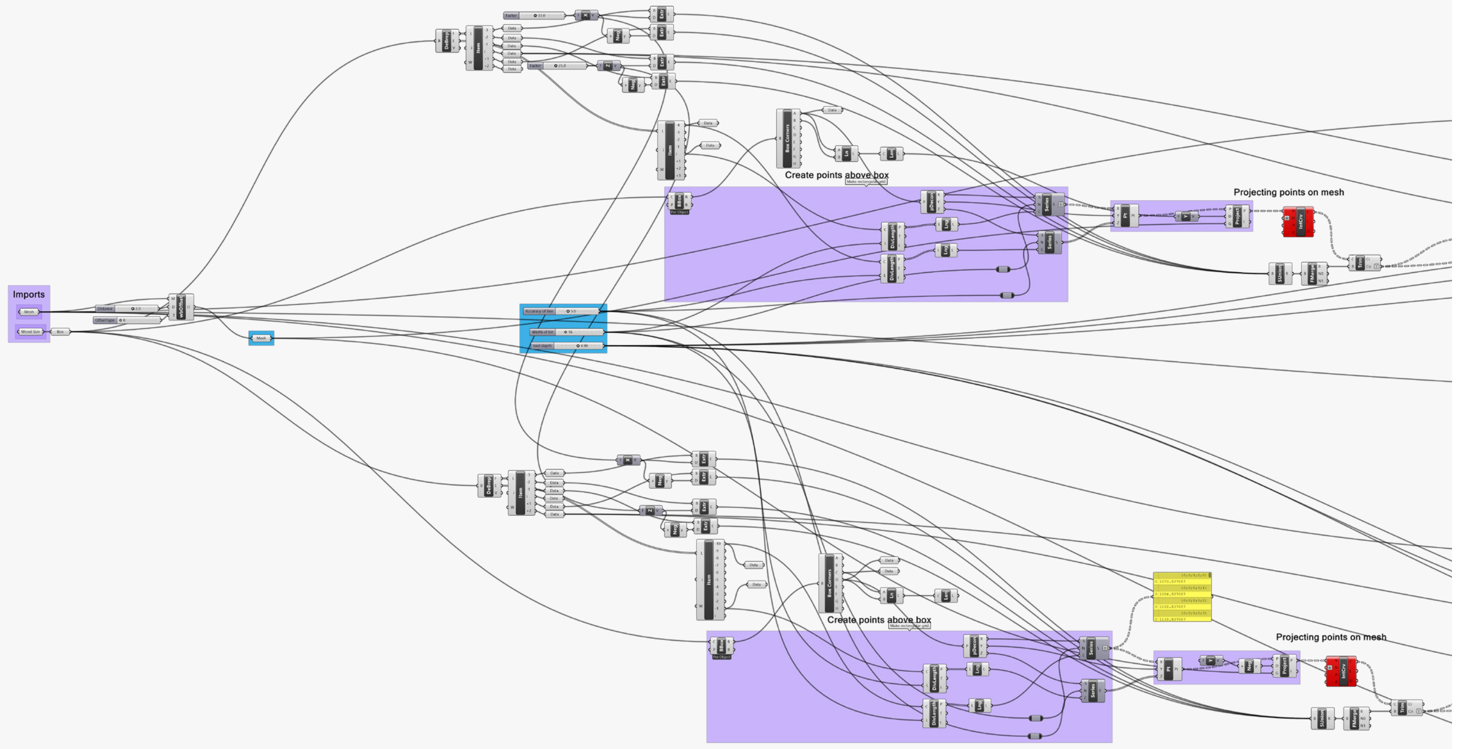
fine removal

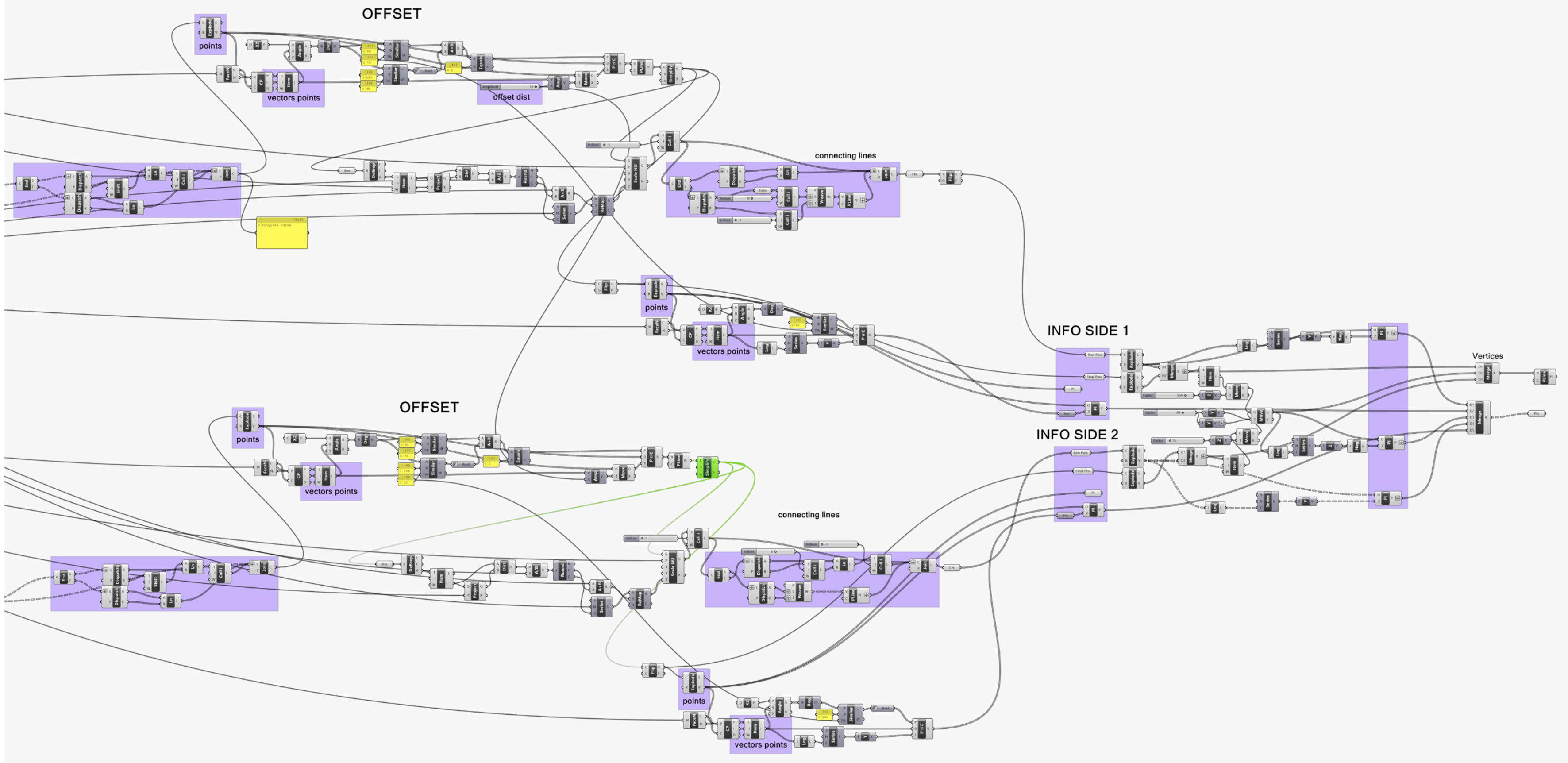




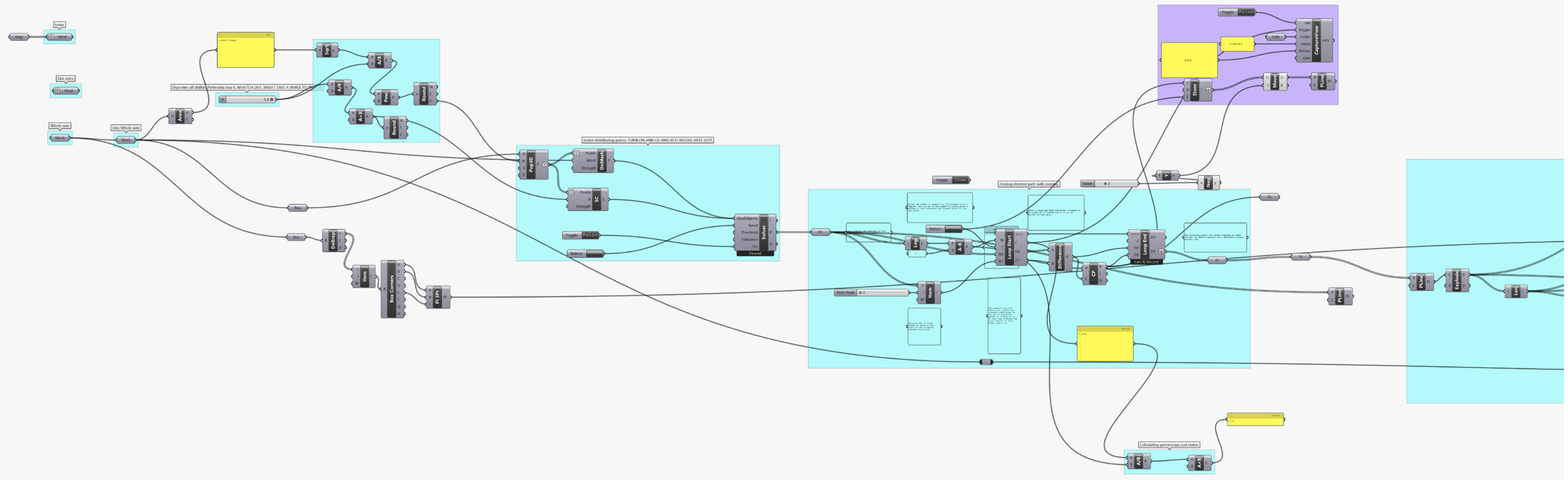
```
202 0.970185
203 0.993343
204 0.997567
205 0.133853
206 0.990396
207 0.155413
208 0.993213
209 0.989176
210 0.98188
211 0.999453
212 0.990778
```

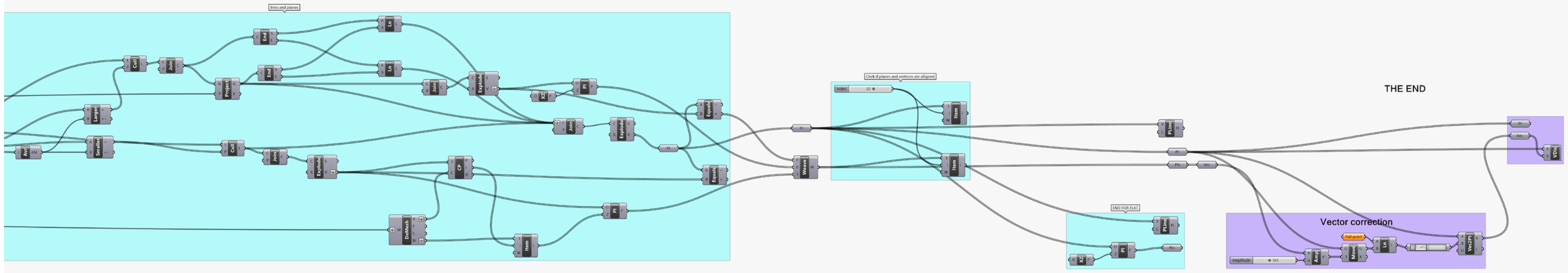
raw removal


Imports
Create points above box
Projecting points on mesh
Create points above box
Projecting points on mesh

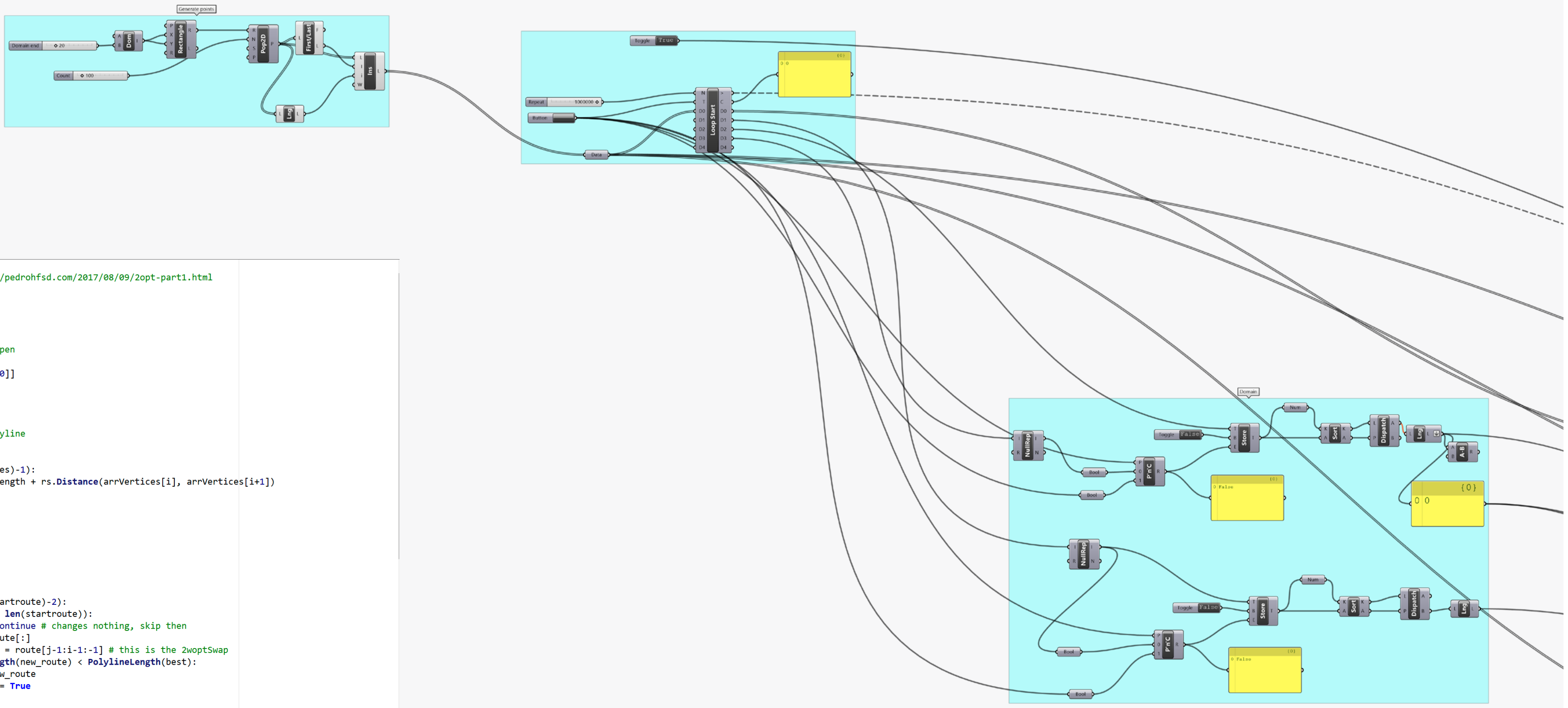

OFFSET
points
vectors points
offset dist
connecting lines
points
vectors points
INFO SIDE 1
Vertices
OFFSET
points
vectors points
connecting lines
INFO SIDE 2
points
vectors points

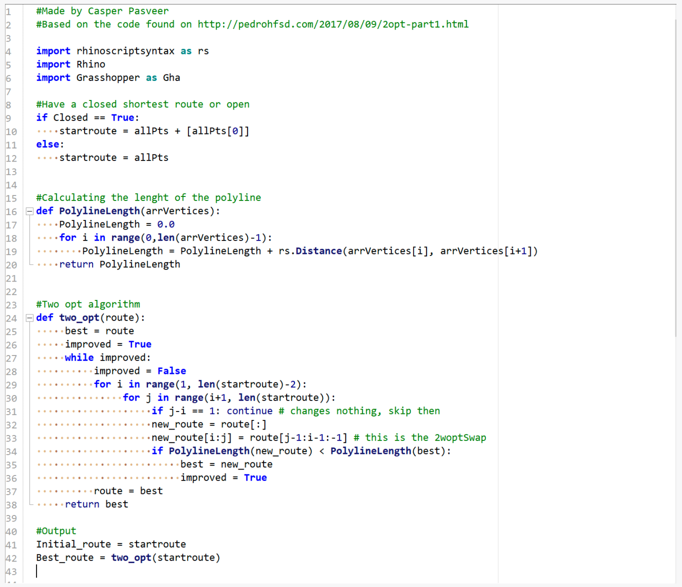
travelling salesman


Sides
Flat sides
Whole side
One Whole side
Diameter off drills. Preferably less 6. WHATCH OUT, MIGHT TAKE A WHILE TO
Solver distributing points. TURN ON AND OF AND REST BEFORE NEXT STEP
Finding shortest path with looping
Calculating percentage, just status
Pop3D
OcMesh
Solver
Loop Start
Loop End
Difference
CaptureView
Box Corners
Toggle
Button

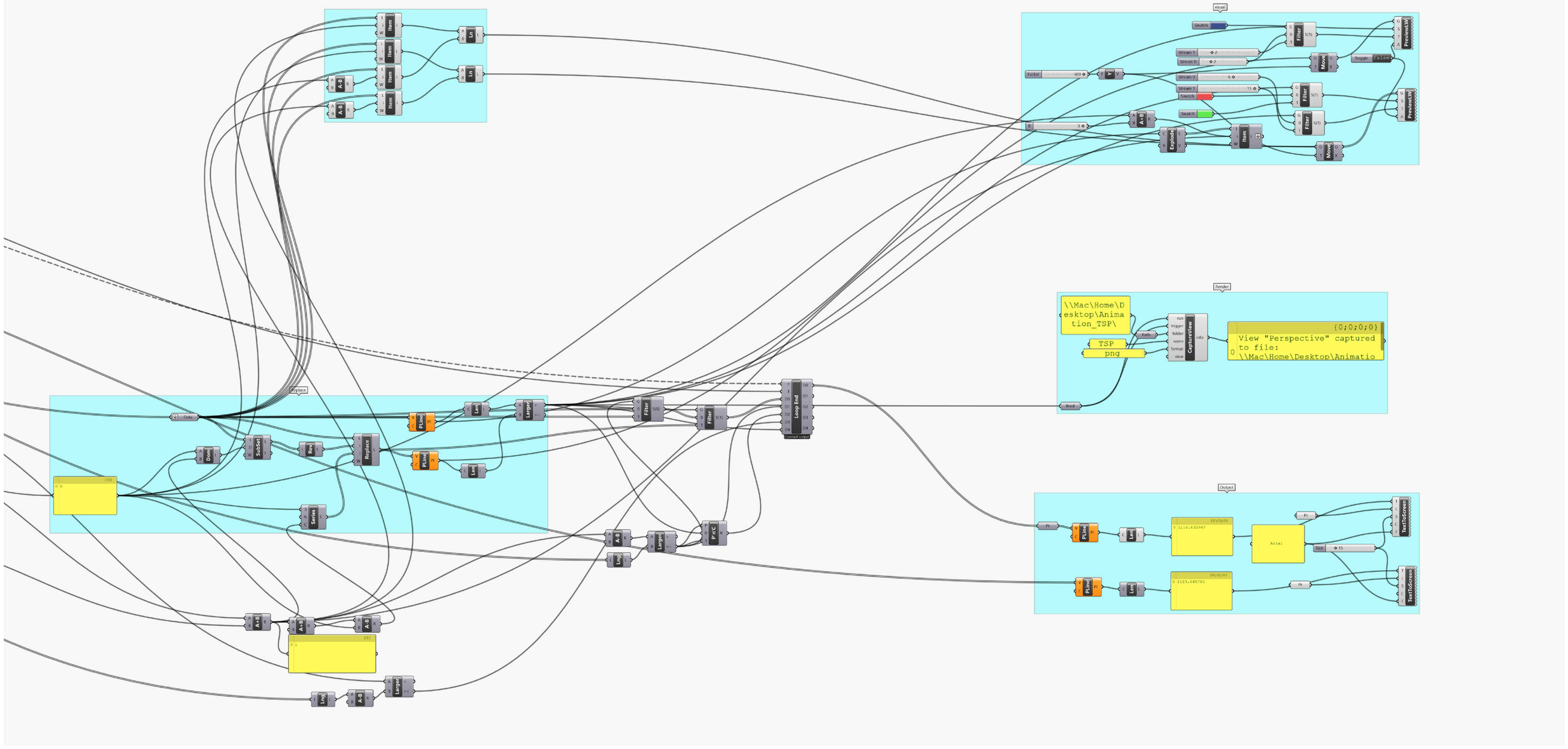

lines and planes
Chek if planes and verteces are alligned
Index 22
END FOR FLAT
THE END
Vector correction
Amplitude

2 - opt





```python
#Made by Casper Pasveer
#Based on the code found on http://pedrohfsd.com/2017/08/09/2opt-part1.html

import rhinoscriptsyntax as rs
import Rhino
import Grasshopper as Gha

#Have a closed shortest route or open
if Closed == True:
    startroute = allPts + [allPts[0]]
else:
    startroute = allPts


#Calculating the lenght of the polyline
def PolylineLength(arrVertices):
    PolylineLength = 0.0
    for i in range(0,len(arrVertices)-1):
        PolylineLength = PolylineLength + rs.Distance(arrVertices[i], arrVertices[i+1])
    return PolylineLength


#Two opt algorithm
def two_opt(route):
     best = route
     improved = True
     while improved:
          improved = False
          for i in range(1, len(startroute)-2):
              for j in range(i+1, len(startroute)):
                  if j-i == 1: continue # changes nothing, skip then
                  new_route = route[:]
                  new_route[i:j] = route[j-1:i-1:-1] # this is the 2woptSwap
                  if PolylineLength(new_route) < PolylineLength(best):
                      best = new_route
                      improved = True
          route = best
     return best

#Output
Initial_route = startroute
Best_route = two_opt(startroute)
```



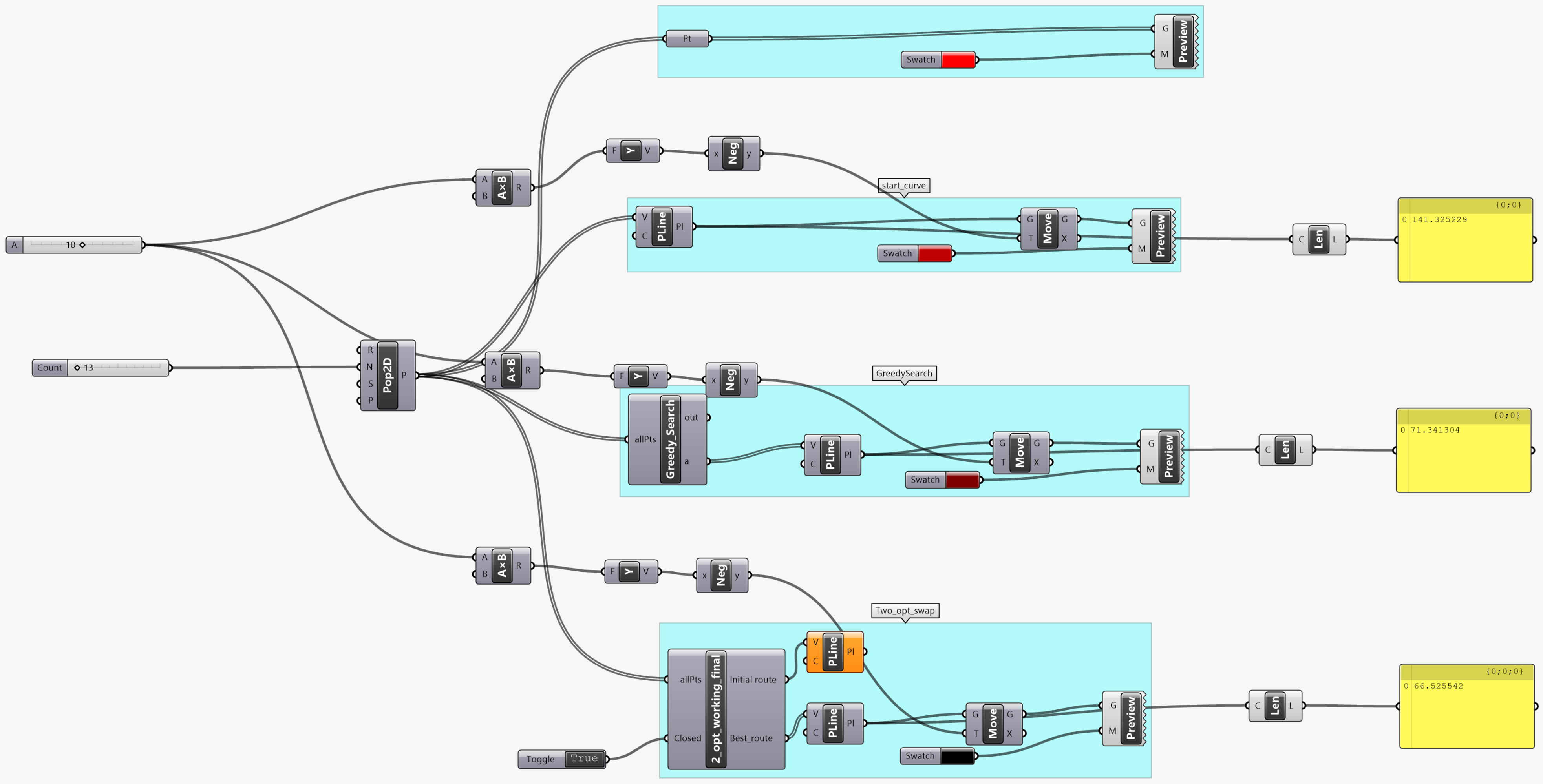
```
\\Mac\Home\Desktop\Animation_TSP\
```

```
{0;0;0;0}
0 View "Perspective" captured
  to file:
  \\Mac\Home\Desktop\Animatio
```

```
{0;0;0}
0 1116.632947
```

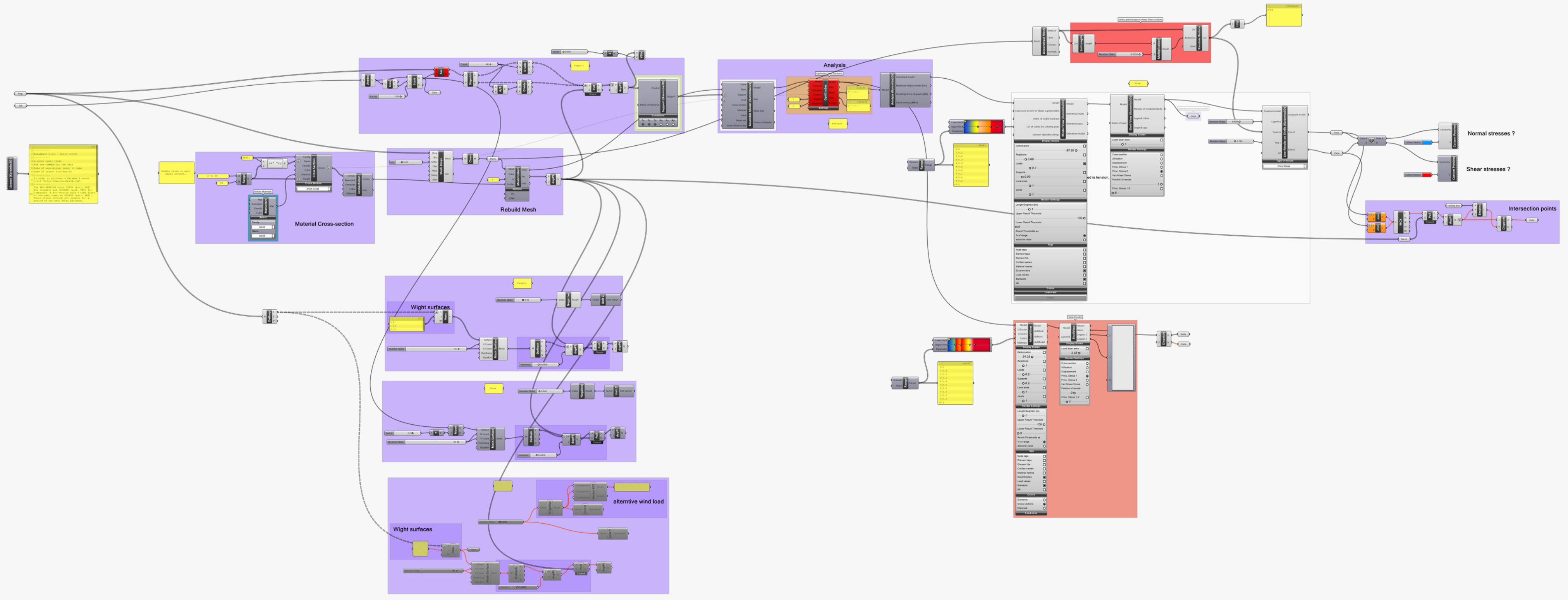
```
{0;0;0}
0 1123.645765
```

greedy line

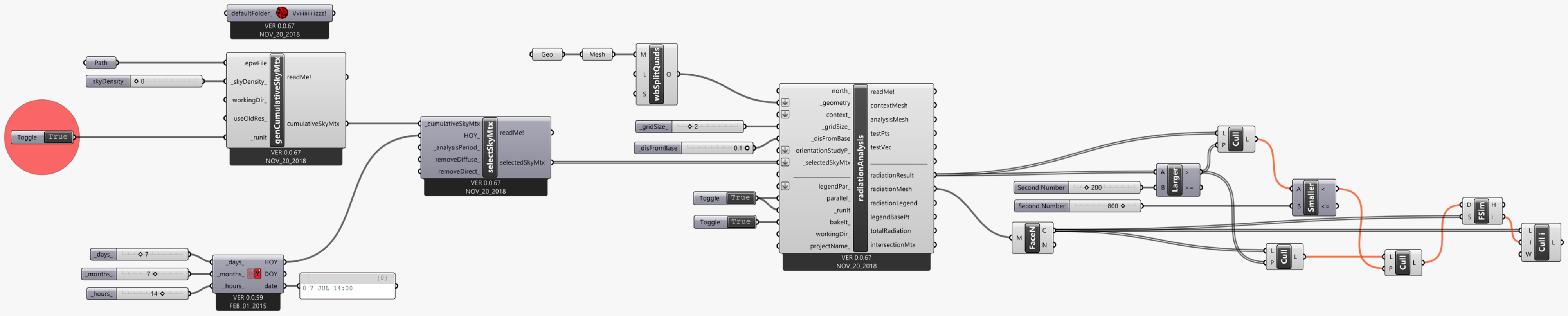


A 10
Count 13
Pop2D
A×B
Pt
Preview
Swatch
start_curve
PLine
Move
Neg
Len
{0;0}
0 141.325229
GreedySearch
Greedy_Search
allPts
out
a
{0;0}
0 71.341304
Two_opt_swap
2_opt_working_final
Initial route
Best_route
Closed
Toggle True
{0;0;0}
0 66.525542

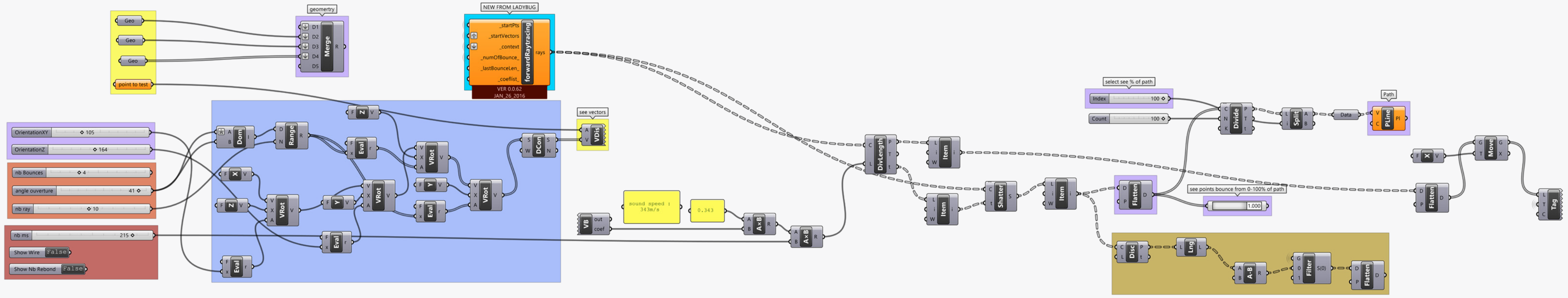
stress analysis



radiation analysis



acoustic analysis


geomertry
NEW FROM LADYBUG
forwardRaytracing
see vectors
point to test
OrientationXY 105
OrientationZ 164
nb Bounces 4
angle ouverture 41
nb ray 10
nb ms 215
Show Wire False
Show Nb Rebond False
sound speed : 343m/s
0.343
select see % of path
Index 100
Count 100
Path
see points bounce from 0-100% of path
1.000