

File Browser

Python_codes

- posdata_folder
- __pycache__
- 3ns.txt
- 4adjm.txt
- 4ns.txt
- 6adjm.txt
- 6adjm-classic.txt
- 6ns.txt
- BeautifuliedGWO.py
- BeautifuliedGWO1.py
- BeautifuliedGWO2.py
- BeautifuliedGWO3.py
- BeautifuliedGWO4.py
- BeautifuliedGWO5.py
- BeautifuliedGWO6.py
- BeautifuliedGWO7.py
- BeautifuliedGWO8.py
- BeautifuliedNitin.py
- BeautifuliedNitin1.py
- BeautifuliedNitin2.py
- clusteredcomputing (1).py
- ClusteredComputing.py
- clusteredcomputingGWO
- ClusteredComputinGWO1.py
- ez_domain_wall.py
- interaction.txt
- penalty.txt
- posdata.txt
- posdata0.txt
- posdata1.txt
- posdata2.txt
- posdata3.txt
- posdata4.txt
- posdata5.txt
- quantumrouting-example.cc
- quantumrouting-example-2python.cc
- quantumrouting-example-clustered.cc
- quantumrouting-example-simple.cc
- quantumrouting-fqst.cc
- SanityCplusplusCheck.py
- SolvBund.py
- SolveBundleCalls.py

BeautifuliedGWO8.py

```

25 def segGraph(
26     segNo,
27     segAmount,
28     theta_core_pos,
29     ang,
30 ):
31
32     max_graph_size = 6
33     radius = 50
34     theta = 0
35     ang_seg = 360 / segAmount
36
37     pos_matrix = numpy.zeros((2, max_graph_size + 1))
38     adjacency_matrix = numpy.zeros((max_graph_size + 1, max_graph_size
39                                     + 1))
40
41     # pos_matrix[0, 0] = theta_core_pos[0, segNo]
42     # pos_matrix[1, 0] = theta_core_pos[1, segNo]
43
44     centr_x = theta_core_pos[0, segNo]
45     centr_y = theta_core_pos[1, segNo]
46
47     dist = radius
48     min_indx = 0
49     temp_dist = radius
50     pi = 3.1415926
51     for i in range(1, max_graph_size + 1):
52         r_radius = random.random()
53         temp_radius = r_radius * radius
54
55         r_angular = random.random()
56         temp_angular = (r_angular * ang_seg + ang - ang_seg) * 2 * pi \
57                       / 360
58
59         x = temp_radius * math.cos(temp_angular)
60         y = temp_radius * math.sin(temp_angular)
61
62         temp_dist = math.sqrt((x - centr_x) * (x - centr_x) + (y
63                               - centr_y) * (y - centr_y))
64
65         if temp_dist <= dist:
66             dist = temp_dist
67             min_indx = i
68
69         pos_matrix[0, i] = x
70         pos_matrix[1, i] = y
71
72     temp_x = pos_matrix[0, 0]
73     temp_y = pos_matrix[1, 0]
74
75     pos_matrix[0, 0] = pos_matrix[0, min_indx]
76     pos_matrix[1, 0] = pos_matrix[1, min_indx]
77

```

posdata0.txt

Python Tab Width: 8 Ln 89, Col 1