Timer unit: 1e-06 s

Total time: 0.179038 s
File: <ipython-input-38-507332c88754>
Function: main at line 25

Line #	Hits	Time	Per Hit	% Time	Line Contents
25					def main(n, alpha, T):
26					"""
27					n: there are n points in the unit square
28					alpha: alpha is the portion of points you redistribute at every time step
29					T: number of iterations
30					return:
31					length_array: list of caluclated values L(0), L(1),\dots, L(T-1)
32					plot the diagram L(t) for t= 0, 1,...,T-1
33					"""
34	1	37.0	37.0	0.0	x = np.random.uniform(size=n)
35	1	19.0	19.0	0.0	y = np.random.uniform(size=n)
36	1	18.0	18.0	0.0	positions = list(zip(x,y))
37	1	2.0	2.0	0.0	length_array = []
38	1	16.0	16.0	0.0	position_status = dict((key, value) for (key, value) in zip(positions, [False]*(len(positions)
39	1	12.0	12.0	0.0	starting_point = random.choice(positions)
40	1	4.0	4.0	0.0	positions.remove(starting_point)
41	1	2.0	2.0	0.0	position_status[starting_point] = True
42					
43					
44	6	10.0	1.7	0.0	for t in range(0,T):
45					#perturbation of points
46	5	5.0	1.0	0.0	if t > 0:
47	4	12.0	3.0	0.0	positions = list(position_status.keys())
48	4	8.0	2.0	0.0	perturb_no = n*alpha
49	4	13770.0	3442.5	7.7	new_pos, old_pos = returnPerturbations(perturb_no, positions)
50	4	50.0	12.5	0.0	positions = [rp for rp in positions if rp not in old_pos]
51	4	9.0	2.2	0.0	positions.extend(new_pos)
52	4	69.0	17.2	0.0	position_status = {key: position_status[key] for key in list(position_status.keys()) i
53					
54	55	90.0	1.6	0.1	for p in range(0,n):
55	50	77.0	1.5	0.0	if len(positions) > 1:
56	48	61.0	1.3	0.0	path_length = 0
57	48	70.0	1.5	0.0	if p == 0:
58	5	9.0	1.8	0.0	previous_point = starting_point
59					
60					#find shortest path through all points
61					#current_point,increment = minEucDistance(previous_point,positions)
62	48	767.0	16.0	0.4	distances = [math.sqrt((previous_point[0]-point[0])**2 + (previous_point[1] - poin
63	48	229.0	4.8	0.1	dist_dict = dict(zip(positions,distances))
64	48	214.0	4.5	0.1	current_point = min(dist_dict, key=dist_dict.get)
65	48	192.0	4.0	0.1	path_length += dist_dict[min(dist_dict, key=dist_dict.get)]
66					
67					#path_length += increment
68	48	102.0	2.1	0.1	positions.remove(current_point)
69	48	80.0	1.7	0.0	position_status[current_point] = True
70	48	63.0	1.3	0.0	previous_point = current_point
71					
72	5	8.0	1.6	0.0	length_array.append(path_length)
73					
74	1	27150.0	27150.0	15.2	plt.xlabel('Iteration (t)')
75	1	36.0	36.0	0.0	plt.ylabel('L(t)')
76	1	272.0	272.0	0.2	plt.title('Mypoic Distance Calculation')
77	1	8075.0	8075.0	4.5	plt.bar([i for i in range(0,T)],length_array)
78	1	127497.0	127497.0	71.2	plt.show()
79					
80	1	3.0	3.0	0.0	return length_array