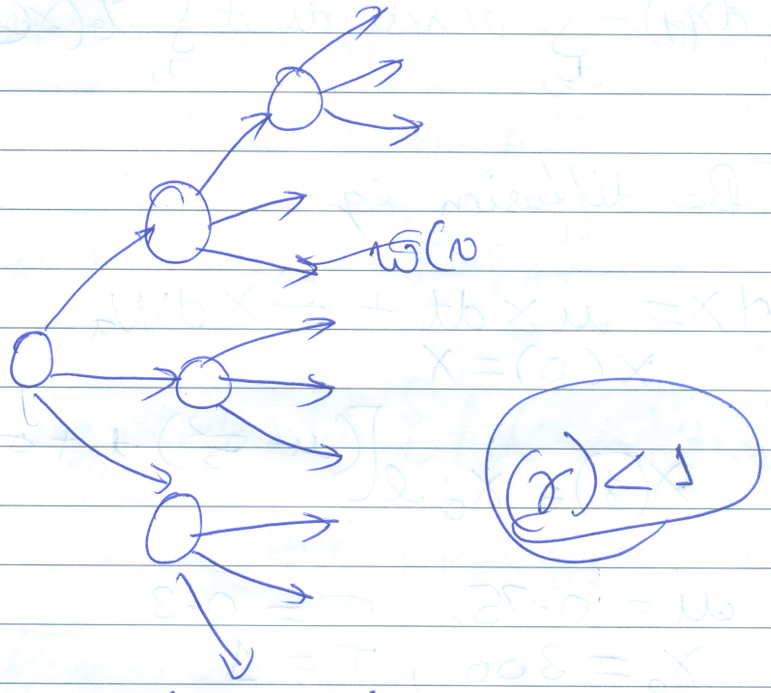
$$(avg) = \left(\gamma \sum_{n \in N}\right)^{\omega}$$



(1) maintain local table list and check it first if there is some issue ⇒ avoid memory lock as much possible.

start with max weight on each node
↳ (to explore new area)

$p_e = \left(\text{phere}\right) \rightarrow$ (if robot position does not change)

~~a node is as~~

→ measure the centrality of the node in exploration ⇒ more central a node, more valuable it is
→ a node is as central as the combined centrality of surrounding node
→ initially all nodes are central equally.

Caution → weights may explode after a few iteration → [(v') has robot at some position as (v)]

(a) $w(v) = \gamma \sum_{v \in N(v)} w(v) + \left(\frac{1}{\gamma}\right) \sum_{v \in N(v)} w(v)$

w has a range [min, max]

if $\gamma \approx 1$ → it will turn like random walk

(b) (v) has robot at different position

$(\Delta\tau) \propto \frac{1}{(\text{path length till that point})}$ ⇒ prefer shorter paths over longer

(1-ρ) → evaporation rate

commit to

promote discovery of new paths → only the discovered nodes will suffer from this evaporation rule.

Iteration step size

(1)	100	(1) 100
(2)	500 (or)	200
(3)	1000	300
(4)	5,000	400
(5)	10,000	500
(6)	50,000	600
(7)	1,00,000	700
		800
		900
		1000
		1100
		1200

N_i → no. of steps in (i)th iter

$(\Delta N_i) = N_{(i+1)} - N_i$

$\Delta N_i \propto \frac{1}{\text{no. new nodes discovered}}$

Logic → if less node discovered ⇒ increase by more.

or better,

$(\Delta\tau) \propto \frac{1}{(\text{cost to that point})}$

$(\Delta\tau) \propto (\text{path weight})$

(path weight) ∝ (no. of node visited by robot) cluster

(path weight) ∝ $\frac{1}{\text{total length of path}}$

γ → retention rate