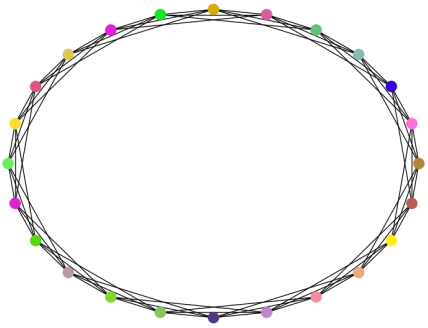
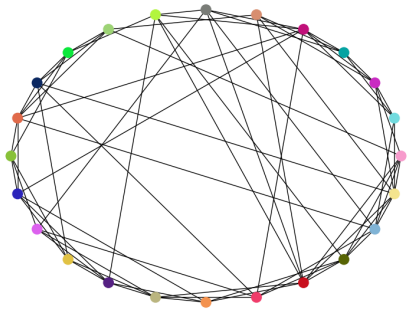
Genetic Algorithm for the Dynamic Shortest Path Routing Problem in Random Networks with Small-World and Scale-Free Properties [Thomas Walsh]

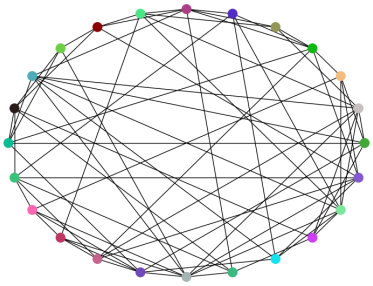
University of Kent

P = 0

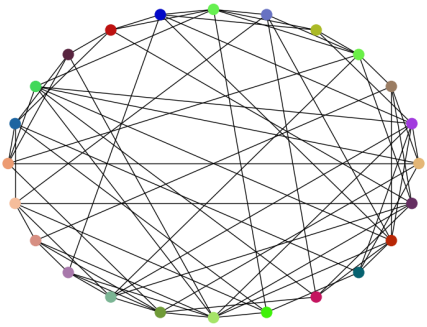
P = 0.5



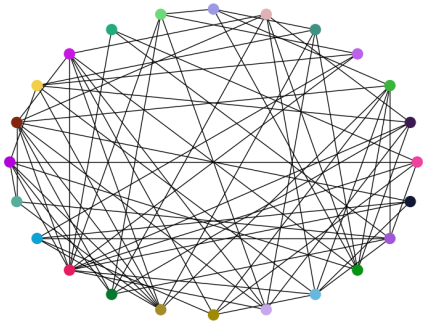




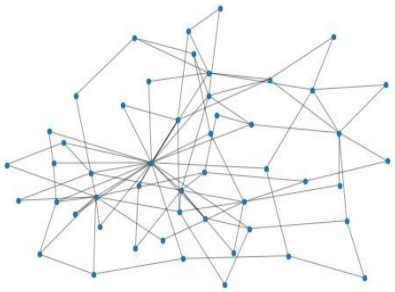
P = 0.75

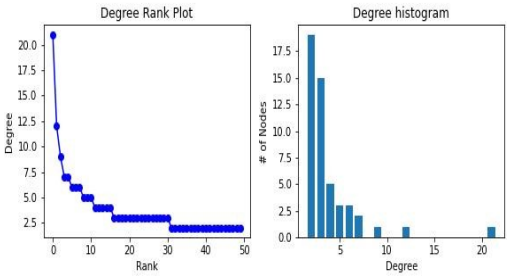


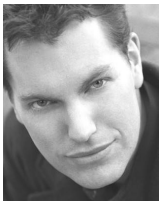
P = 1.0

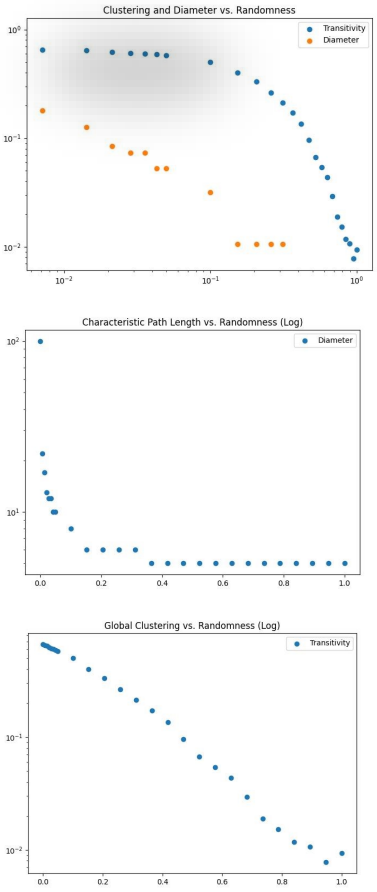


Scale Free Networks: Being Lonely is Normal









$t := 0$ and $t_M := rand(5, 10)$
initialize population $P(0)$ and memory $M(0)$ randomly
repeat
  evaluate population $P(t)$ and memory $M(t)$
  replace the worst individual in $P(t)$ by the elite $E(t-1)$ from $P(t-1)$

  if *change detected* **then**
    $P'(t) :=$retrieveBestMembersFrom$(P(t), M(t))$
  else $P'(t) := P(t)$

  // time to update memory
  if $t = t_M$ || *change detected* **then**
    if $t = t_M$ **then**
      $B_P(t) :=$ retrieveBestMemberFrom$(P'(t))$
    if *change detected* **then** $B_P(t) := E(t-1)$
    if *still any random point in memory* **then**
      replace a random point in memory with $B_P(t)$
    else // replace the most similar memory point
      if $t = t_M$ **then**
        find the memory point $C_M(t)$ closest to $B_P(t)$
        if $f(B_P(t)) > f(C_M(t))$ **then**
          $C_M(t) := B_P(t)$
      if *change detected* **then**
        find the memory point $C_M(t-1)$ closest to $B_P(t)$
        if $f(B_P(t)) > f(C_M(t-1))$ **then**
          $C_M(t-1) := B_P(t)$
    $t_M := t + rand(5, 10)$

  // standard genetic operations
  $P''(t) :=$ selectForReproduction$(P'(t))$
  crossover$(P''(t), p_c)$ // p_c *is the crossover probability*
  mutate$(P''(t), p_m)$ // p_m *is the mutation probability*
  $P(t+1) := P''(t)$
until the termination condition is met // *e.g.,* $t > t_{max}$