```
24-06-04 Ma 1
  Tuesday, June 4, 2024
                        9:41 AM
          https://drive.google.com/file/d/16pZJDXw-4JUlKbZXpAGdH8bHX9gHt5uD/view?usp=drivesdk
                             HC para TSP con Alguna Mejora
                             1. Representación: Xi la i-ésima ciudad visitada
                             2. Función de Evaluación: Largo del tour
                             3. Movimiento: Intercambio entre dos variables
                             4. Criterio de selección de las dos variables: Aleatorio
                             5. Criterio de aceptación: alguna mejora
                             6. Criterio de Término: Máximo número de iteraciones o estancamiento en óptimo
                               local
                             7. Número de re-starts: 5
                             8. Solución Inicial: Aleatoria
                           Procedure HC-TSP-AM
                           Begin
                           For i=1 the num-restarts
                              Solucion-actual = Generar-solucion-inicial
                              While(iteraciones<Maxiter and no-optimo-local)
                                  Var_1 = random(1, ciudades)
                                  Var_2 = random(1, ciudades)
                                  Candidata-solucion-Swap (Solucion-actual, Var1, Var2)
                                  If F(Candidata-solucion) < F(Solucion-actual) then
                                    Solucion-actual=Candidata-solucion
                                  iteraciones++
                              EndWhile
                           EndFor
                           End Procedure
                                              Figure 1: HC-TSP-AM
                               HC para TSP con Mejor Mejora
                              1. Representación: Xi la i-ésima ciudad visitada
                              2. Función de Evaluación: Largo del tour
                              3. Movimiento: Intercambio entre dos variables
                              4. Criterio de selección de una variable: Aleatorio
                              5. Criterio de aceptación: mejor mejora
                              6. Criterio de Término: Máximo número de iteraciones o estancamiento en óptimo
                                local
                              7. Número de re-starts: 5
                              8. Solución Inicial: Aleatoria
                               HC para TSP con Alguna Mejora Representa-
                                cion binaria - Infactibles
                              1. Representación: Xi vale 1 si la ciudad i es visitada en el j-ésimo lugar, 0 sino
                              2. Función de Evaluación: Largo del tour + castigo
                              3. Movimiento: cambio de 0 por 1, cambio de 1 por 0
                              4. Criterio de selección de la variables: Aleatorio
                              5. Criterio de aceptación: alguna mejora
                              6. Criterio de Término: Máximo número de iteraciones o estancamiento en óptimo
                              7. Número de re-starts: 5
                               8. Solución Inicial: Aleatoria
                      Procedure HC-TSP-MM
                      Begin
                      For i=1 the num-restarts
                         Solucion-actual = Generar-solucion-inicial
                         While(iteraciones<Maxiter and no-optimo-local)
                             Var_1 = random(1, ciudades)
                             For k=1 to ciudades
                               if k \neq Var_1 then Var_2 = k
                                Candidata-solucion=Swap(Solucion-actual, Var1, Var2)
                               If F(Candidata-solucion) < F(Solucion-actual) then
                                 Solucion-actual=Candidata-solucion
                             iteraciones++
                         EndWhile
                      EndFor
                      End Procedure
                                          Figure 2: HC-TSP-MM
                      Procedure HC-TSP-RB
                      Begin
                      For i=1 the num-restarts
                         Solucion-actual = Generar-solucion-inicial
                         While(iteraciones<Maxiter and no-optimo-local)
                             Var_1 = random(1, ciudades)
                             Var_2 = random(1, ciudades)
                             Candidata-solucion=Cambiar-valor(Solucion-actual, X_{Var_1Var_2})
                             If F(Candidata-solucion) < F(Solucion-actual) then
                                Solucion-actual=Candidata-solucion
                             iteraciones++
                         EndWhile
                       EndFor
                       End Procedure
                                          Figure 3: HC-TSP-RB
                          Un Greedy para TSP

    Representación: X<sub>i</sub> la i-ésima ciudad visitada

                         2. Regla de Transición: La ciudad permitida más cercana
                       Procedure Greedy-1-TSP
                       Begin
                       X_1 = 1
                       i=1
                       Ciudades-Permitidas=(2,3,..ciudades)
                       Whilehay Ciudades-Permitidas
                          For j=2 to ciudades
                             If j \in Ciudades-permitidas then
                              X_{i+1}=Ciudad-Min(d_{X_{i,j}})
                           1++
                          Ciudades-Permitidas-Ciudades-Permitidas-X_i
                       EndWhile
                       End Procedure
                                         Figure 4. Un Greedy TSP
Greedy (constructivo)

Representación

f. de evaluación

Constructivo
     • f. miope
• pto inicial
• # ptos iniciales a Intentar & parámetros
         exploración = ptos iniciales
         explotación = cuando te mudos a una sol mor.
            Hill Climbing (reparasor)
     · Representación
     . Je evaluación
     · Sol. inicial (Sp)
     explora = restart
          explota = movimiento
   CUANDO no puedo contro las reestricciones,
    Se puede añadir una componente en la F.O.
                    f(x) - (P·# Ry insatisfechas)
                     no en candilar nos con una sol. Que parece
                   mejor, pero no cumple R/ (una penaliza-
      Cion
      Metaheuristicas a HC.
                                                                               METORAL A HC, el CUAL
   (guardar siempre en memoria el mejorhistorico) de Queda muy rápido
                                                                                estancado en sol.
    Simulated Annealing (AM.)
                                                                              Je inspira en la termo-
pinamica, (metales)
      (On Cierta T° Autorizo A mi Algoritmo A
  Aceptar una Sol peor, para converger a una
   501. mejor A largo plazo.
     CUANDO la T°es muy baja, actua como HC.
    explora = poder aceptar sol. peores are 19 Actual explota = 11 11 mfr
       taboo Search (MM)
     no se fita en la sol. Actual, solo elige el
metor vecino, por etm:
               501 Actual f(x)
-30]
                                        20 - 7 elige este
           pero el Algoritmo 15
no Sabe
    Riesgo de existir ciclos, ir a un vecino Que
Ya revisé.
   7 501: lista taboo (objetivo: evitar ciclos)
     explora = Aceptar sol. peores, restart
explota = elegir sol. mejor Que la actual.
                                                                   PARÁMETROS importantes

CADA cuanto bajar To

Cuanto bajar To
                    Algoritmo Simulated Annealing
                    Paso 0: Inicialización
                          X:=solución inicial factible
                          tmax:= máximo número de iteraciones
                                                                   # iteraciones# restart
                          q:= Temperatura alta inicial
                          Mejor solución:= X
                                                                no conviene bajar muy rapido la
                          número de soluciones =t :=0
                                                                  To para no converger rapido A UN H.C.
                    Paso 1: Parada
                          Si no hay un movimiento posible de la solución actual o si t=tmax entonces
                             parar. Entregar Mejor solución.
                    Paso 2: Posible Movimiento
                          Elegir aleatoriamente algún movimiento factible \triangle x(t+1)
                          Calcular el incremento (o disminución) △objetivo
                    Paso 3: Aceptación
                          Si X(t+1) mejora el objetivo o si P(\exp(\triangle objetivo/q)) \ge random(0,1).
                          Modificar X(t+1) := X(t) + \triangle x(t+1) sino volver al Paso 2.
                    Paso 4: Reemplazar el mejor
                          Si el valor de la función objetivo de X(t+1) es superior a Mejor solución en-
                             tonces a Mejor solución:= X(t+1)
                     Paso 5: reducción de la Temperatura
                          Si ha pasado un número suficiente de iteraciones desde el último cambio de la
                             temperatura, reduzca q.
                     Paso 6: Incrementar
                          t:=t+1, volver a Paso 1.
                Algoritmo Tabu Search:
                Paso 0: Inicialización
                      X:=solución inicial factible
                      tmax:= máximo número de iteraciones
                      Mejor solución:=X
                      número de soluciones = t := 0
                      lista tabu:= vacía
                Paso 1: Parada
                      Si todos los movimientos posibles de la solución actual son tabu o si t=tmax
                         entonces parar. Entregar Mejor solución.
                Paso 2: Mover
                      Aplicar el movimiento no-tabu factible \triangle x(t+1)
                Paso 3: Iteración
                      Modificar X(t+1) := X(t) + \triangle x(t+1)
               Paso 4: Reemplazar el mejor
                      Seleccionar aquel X(t+1) dentro del vecindario con el mejor valor de la
                         función objetivo Mejor solución:= X(t+1)
               Paso 5: Actualizar Lista Tabu
                     Eliminar desde la lista tabu cualquier movimiento que ha permanecido un sufi-
                         ciente número de iteraciones en la lista.
                     Agregar un conjunto de movimientos que involucran un retorno inmediato desde
                         X(t+1) a X(t)
               Paso 6: Incrementar
                     t:=t+1, volver a Paso 1.
```