

PRÁCTICA 2. Recocido Simulado

Alumno: Isaac André Canalizo Mendoza 201728726

I Objetivos

General: Diseñar un algoritmo de recocido simulado para la resolución de dos problemas discretos y uno continuo.

Específico: Para cada uno de los problemas encontrar cuál es la configuración de Simulated Annealing que obtiene en promedio el mejor resultado.

II Desarrollo

Problemas continuos

1) My_func4:

a) Temperatura inicial:

Se estableció con un valor de 100

b) Función de descenso de temperatura:

Se estableció como $((100)/(i+1))$ ya que se obtenía un descenso controlado adecuado para la elección de vecindario y la función de aceptación mostraba resultados favorables.

c) Función de aceptación:

Metrópolis con un valor de $k=10$

d) Configuración del vecindario:

```
for i in range(100):
    temp = 100/(i+1)
    if(temp<10 and temp > 5):
        #El tamaño del vecindario es de 10
        x1, x2 = vecindario(5, x1prim), vecindario(5, x2prim)
    elif(temp<=5 and temp > 2.5):
        #Varia en 4 el tamaño del vecindario
        x1, x2 = vecindario(2, x1prim), vecindario(2, x2prim)
    elif(temp<=2.5 and temp > 1.5):
        #El tamaño del vecindario sigue siendo de 4 pero solo se modifica
        uno de los dos valores
        if( i%2 == 0):
            x1 = vecindario(2, x1prim)
        else:
            x2 = vecindario(2, x2prim)
    elif(temp <= 1.5):
        #El tamaño de vecindario se reduce a 2 y se continua aplicando a un
        solo valor
```

```

    if( i%2 == 0):
        x1 = vecindario(1, x1prim)
    else:
        x2 = vecindario(1, x2prim)
elif(temp <= 1.3):
    #El tamaño del vecindario se reduce a 1
    if( i%2 == 0):
        x1 = vecindario(.5, x1prim)
    else:
        x2 = vecindario(.5, x2prim)
else:
    #Se generan aleatorios en el rango [-25, 25]
    x1, x2 = generateRand(), generateRand()

```

La función vecindario es la siguiente :

```

def vecindario(rango, numero):
    limInferior = numero - rango
    limSuperior = numero + rango

    if(limSuperior > 25.0):
        r = random.uniform(limInferior, 25.0)
    elif(limInferior < -25.0):
        r = random.uniform(-25.0, limSuperior)
    else:
        r = random.uniform(limInferior, limSuperior)
    return r

```

2) My_func5:

a) Se siguió la misma configuración que en el problema anterior

Problema discreto

1) My_graph1:

- a) Temperatura inicial: 500
- b) Función de descenso de temperatura: $temp = 500/(i+1)$
- c) Función de aceptación: metrópolis con $k = 100$
- d) Elección de vecindario

Configuración del vecindario:

```

temp = 500/(i+1)

if(temp<1):

```

```

#Antes de llegar acá itera 500 veces
if(temp>0.67):
    # Itera hasta 750
    if(i%2 == 0):
        r = random.randint(1,7)
        x = switch_pair(r, comb)
    else:
        x = switch_pair(8, comb, 8)
elif(temp>0.51):
    #Itera hasta 980
    r = random.randint(0,2)
    match r:
        case 0:
            x = switch_pair(6, comb)
        case 1:
            x = switch_pair(7, comb)
        case 2:
            x = switch_pair(8, comb, 4)
elif(mejor > 5):
    #Itera las últimas 20 veces
    r = random.randint(0,18)
    x = switch_pair(0, comb, r)

## Cambio de pares
def opc1(lista: List, pos:int):
    lista[pos], lista[pos+1] = lista[pos+1], lista[pos]
    return lista

##### CAMBIO DE EXTREMOS
#####
def opc2(lista):
    for i in range(5):
        lista[i], lista[19-i] = lista[19-i], lista[i]
    return lista

##### CAMBIO DE CENTRO
#####
def opc3(lista):
    for i in range(5):
        lista[5+i], lista[14-i] = lista[14-i], lista[5+i]
    return lista

##### MEZCLA DE PRIMER Y TERCER CUARTO
#####
def opc4(lista):
    for i in range(5):

```

```

        lista[i], lista[14-i] = lista[14-i], lista[i]
    return lista

##### MEZCLA SEGUNDO Y CUARTO CUARTO
#####

def opc5(lista):
    for i in range(5):
        lista[5+i], lista[19-i] = lista[19-i], lista[5+i]
    return lista

##### MEZCLA PRIMERA MITAD
#####

def opc6(lista):
    used = []
    con = 0
    while len(used) < 10:
        r = random.randint(0,9)
        if(r not in used):
            lista[con], lista[r] = lista[r], lista[con]
            used.append(r)
            con += 1
    return lista

##### MEZCLA SEGUNDA MITAD
#####

def opc7(lista):
    used = []
    con = 5
    while len(used) < 10:
        r = random.randint(10,19)
        if(r not in used):
            lista[con], lista[r] = lista[r], lista[con]
            used.append(r)
            con += 1
    return lista

def opc8(lista, numRand):
    rand = []
    while len(rand) < numRand:
        r = random.randint(0,19)
        if(r not in rand):
            rand.append(r)
    lim = int(numRand/2)
    for i in range(lim):

```

```

        lista[rand[i]], lista[ rand[(numRand-1)-i] ] = lista[rand[(numRand-
1)-i]], lista[rand[i]]
    return lista

```

IV Resultados

Para el primer problema continuo obtiene un resultado promedio de: 0.464763 con 10 iteraciones

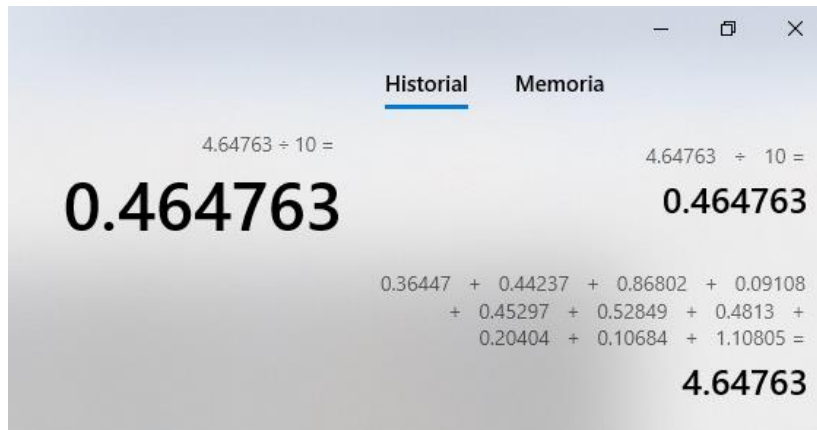


Fig. 1.1 Captura de suma de resultados de diez corridas y su promedio

```

C:\Windows\System32\cmd.exe
I actual: 74 resultado: 24.317 mejor: 1.108 combinación: 17.58253451196999, -11.701765147754474 temp: 1.351
I actual: 75 resultado: 18.264 mejor: 1.108 combinación: 17.784124893865872, -11.701765147754474 temp: 1.333
I actual: 76 resultado: 1.958 mejor: 1.108 combinación: 17.784124893865872, -11.388714930871275 temp: 1.316
I actual: 77 resultado: 26.287 mejor: 1.108 combinación: 16.235179318286672, -11.388714930871275 temp: 1.299
I actual: 78 resultado: 19.63 mejor: 1.108 combinación: 16.235179318286672, -12.067235435060269 temp: 1.282
I actual: 79 resultado: 14.253 mejor: 1.108 combinación: 16.312391826465703, -12.067235435060269 temp: 1.266
I actual: 80 resultado: 5.363 mejor: 1.108 combinación: 16.312391826465703, -11.634594038633212 temp: 1.25
I actual: 81 resultado: 2.874 mejor: 1.108 combinación: 17.023713591817558, -11.634594038633212 temp: 1.235
I actual: 82 resultado: 21.436 mejor: 1.108 combinación: 17.023713591817558, -12.76210105363912 temp: 1.22
I actual: 83 resultado: 22.34 mejor: 1.108 combinación: 16.38808306423011, -12.76210105363912 temp: 1.205
I actual: 84 resultado: 23.296 mejor: 1.108 combinación: 16.38808306423011, -11.319374135612392 temp: 1.19
I actual: 85 resultado: 8.627 mejor: 1.108 combinación: 18.093362559042756, -11.319374135612392 temp: 1.176
I actual: 86 resultado: 23.666 mejor: 1.108 combinación: 18.093362559042756, -12.192290617761227 temp: 1.163
I actual: 87 resultado: 1.359 mejor: 1.108 combinación: 17.151154831218385, -12.192290617761227 temp: 1.149
I actual: 88 resultado: 26.346 mejor: 1.108 combinación: 17.151154831218385, -12.522690472699507 temp: 1.136
I actual: 89 resultado: 42.974 mejor: 1.108 combinación: 17.55607840116564, -12.522690472699507 temp: 1.124
I actual: 90 resultado: 46.112 mejor: 1.108 combinación: 17.55607840116564, -12.577064631788224 temp: 1.111
I actual: 91 resultado: 7.948 mejor: 1.108 combinación: 16.19692723057595, -12.577064631788224 temp: 1.099
I actual: 92 resultado: 7.288 mejor: 1.108 combinación: 16.19692723057595, -12.564056660379979 temp: 1.087
I actual: 93 resultado: 16.578 mejor: 1.108 combinación: 16.997488849504713, -12.564056660379979 temp: 1.075
I actual: 94 resultado: 13.502 mejor: 1.108 combinación: 16.997488849504713, -12.077569189409214 temp: 1.064
I actual: 95 resultado: 14.427 mejor: 1.108 combinación: 16.658955877171447, -12.077569189409214 temp: 1.053
I actual: 96 resultado: 13.153 mejor: 1.108 combinación: 16.658955877171447, -12.935232523458623 temp: 1.042
I actual: 97 resultado: 11.013 mejor: 1.108 combinación: 16.719788987427734, -12.935232523458623 temp: 1.031
I actual: 98 resultado: 16.873 mejor: 1.108 combinación: 16.719788987427734, -12.660025071180062 temp: 1.02
I actual: 99 resultado: 49.896 mejor: 1.108 combinación: 17.468787937474964, -12.660025071180062 temp: 1.01
I actual: 100 resultado: 16.568 mejor: 1.108 combinación: 17.468787937474964, -12.10044750006666 temp: 1.0
La mejor combinación es: 17.09507220917006, -12.238418618541544 y su puntaje es 1.10805
C:\Users\waffl\Documents\PRIMAVERA 2023\Tec. de inteligencia artificial\Practica2\Continuos>

```

Fig. 1.2 Resultado de la última iteración

Para el segundo problema continuo obtiene un resultado promedio de: 0.565086 con 10 iteraciones

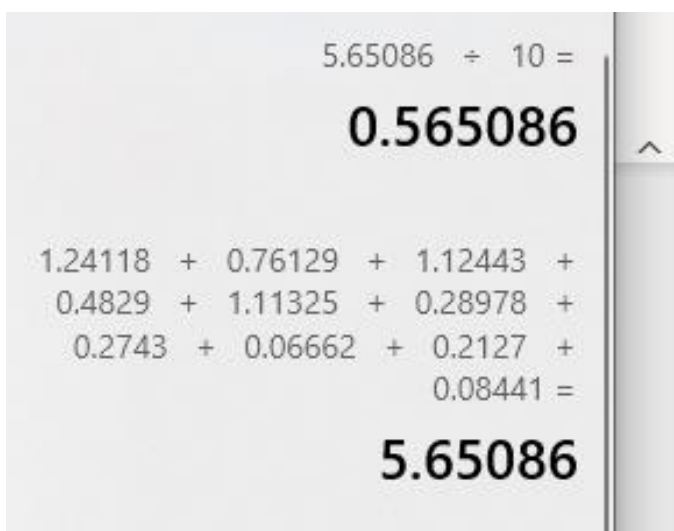


Fig. 1.1 Captura de suma de resultados de diez corridas y su promedio

```
C:\Windows\System32\cmd.exe
I actual: 58 resultado: 44.034 mejor: 3.58 combinación: 9.412383981660014, -18.128969514135616 temp: 1.724
I actual: 59 resultado: 48.852 mejor: 3.58 combinación: 11.050822161773134, -18.128969514135616 temp: 1.695
I actual: 60 resultado: 18.644 mejor: 3.58 combinación: 11.050822161773134, -16.69348376364634 temp: 1.667
I actual: 61 resultado: 0.499 mejor: 3.58 combinación: 9.903569938569495, -16.69348376364634 temp: 1.639
I actual: 62 resultado: 32.488 mejor: 0.499 combinación: 9.903569938569495, -15.726872368521125 temp: 1.613
I actual: 63 resultado: 6.972 mejor: 0.499 combinación: 10.36630509321811, -15.726872368521125 temp: 1.587
I actual: 64 resultado: 31.849 mejor: 0.499 combinación: 10.36630509321811, -15.323057130229822 temp: 1.562
I actual: 65 resultado: 30.989 mejor: 0.499 combinación: 10.454897097506578, -15.323057130229822 temp: 1.538
I actual: 66 resultado: 79.077 mejor: 0.499 combinación: 10.454897097506578, -18.42464720585354 temp: 1.515
I actual: 67 resultado: 67.073 mejor: 0.499 combinación: 9.438734435898104, -18.42464720585354 temp: 1.493
I actual: 68 resultado: 13.465 mejor: 0.499 combinación: 9.438734435898104, -16.789530205988918 temp: 1.471
I actual: 69 resultado: 13.397 mejor: 0.499 combinación: 9.409840434140044, -16.789530205988918 temp: 1.449
I actual: 70 resultado: 10.07 mejor: 0.499 combinación: 9.409840434140044, -16.916196849878425 temp: 1.429
I actual: 71 resultado: 0.084 mejor: 0.499 combinación: 10.022185746094792, -16.916196849878425 temp: 1.408
I actual: 72 resultado: 37.474 mejor: 0.084 combinación: 10.022185746094792, -17.79392292157922 temp: 1.389
I actual: 73 resultado: 47.639 mejor: 0.084 combinación: 9.41115608892232, -17.79392292157922 temp: 1.37
I actual: 74 resultado: 24.824 mejor: 0.084 combinación: 9.41115608892232, -16.02203012445325 temp: 1.351
I actual: 75 resultado: 39.586 mejor: 0.084 combinación: 9.124826573602308, -16.02203012445325 temp: 1.333
I actual: 76 resultado: 33.527 mejor: 0.084 combinación: 9.124826573602308, -16.23483082340733 temp: 1.316
I actual: 77 resultado: 30.018 mejor: 0.084 combinación: 9.172842601593324, -16.23483082340733 temp: 1.299
I actual: 78 resultado: 1.408 mejor: 0.084 combinación: 9.172842601593324, -16.82438554873435 temp: 1.282
I actual: 79 resultado: 23.935 mejor: 0.084 combinación: 10.369064748319431, -16.82438554873435 temp: 1.266
I actual: 80 resultado: 12.879 mejor: 0.084 combinación: 10.369064748319431, -16.465567250872176 temp: 1.25
I actual: 81 resultado: 11.218 mejor: 0.084 combinación: 9.775099309609056, -16.465567250872176 temp: 1.235
I actual: 82 resultado: 37.8 mejor: 0.084 combinación: 9.775099309609056, -16.060746891125394 temp: 1.22
I actual: 83 resultado: 25.851 mejor: 0.084 combinación: 10.926499899354475, -16.060746891125394 temp: 1.205
I actual: 84 resultado: 46.957 mejor: 0.084 combinación: 10.926499899354475, -17.687021485529783 temp: 1.19
I actual: 85 resultado: 54.762 mejor: 0.084 combinación: 10.617294225771534, -17.687021485529783 temp: 1.176
I actual: 86 resultado: 17.034 mejor: 0.084 combinación: 10.617294225771534, -16.12806996377039 temp: 1.163
I actual: 87 resultado: 33.776 mejor: 0.084 combinación: 10.02846083049071, -16.12806996377039 temp: 1.149
I actual: 88 resultado: 4.163 mejor: 0.084 combinación: 10.02846083049071, -16.740268422551125 temp: 1.136
I actual: 89 resultado: 14.044 mejor: 0.084 combinación: 9.433603495579938, -16.740268422551125 temp: 1.124
I actual: 90 resultado: 9.05 mejor: 0.084 combinación: 9.433603495579938, -16.95834845200313 temp: 1.111
I actual: 91 resultado: 9.83 mejor: 0.084 combinación: 10.748425849569076, -16.95834845200313 temp: 1.099
I actual: 92 resultado: 3.235 mejor: 0.084 combinación: 10.748425849569076, -16.47450412202563 temp: 1.087
I actual: 93 resultado: 3.417 mejor: 0.084 combinación: 10.982127340529221, -16.47450412202563 temp: 1.075
I actual: 94 resultado: 13.914 mejor: 0.084 combinación: 10.982127340529221, -16.64761266664922 temp: 1.064
I actual: 95 resultado: 11.725 mejor: 0.084 combinación: 10.913007084534039, -16.64761266664922 temp: 1.053
I actual: 96 resultado: 10.417 mejor: 0.084 combinación: 10.913007084534039, -17.18890733784628 temp: 1.042
I actual: 97 resultado: 21.328 mejor: 0.084 combinación: 10.32464457274267, -17.18890733784628 temp: 1.031
I actual: 98 resultado: 13.908 mejor: 0.084 combinación: 10.32464457274267, -16.13812456525335 temp: 1.02
I actual: 99 resultado: 23.407 mejor: 0.084 combinación: 10.764904925837868, -16.13812456525335 temp: 1.01
I actual: 100 resultado: 10.959 mejor: 0.084 combinación: 10.764904925837868, -16.593006039523747 temp: 1.0
La mejor combinación es: 10.022185746094792, -16.916196849878425 y su puntaje es 0.08441
C:\Users\waffl\Documents\PRIMAVERA 2023\Tec. de inteligencia artificial\Practica2\Continuos>
```

Fig. 1.2 Resultado de la última iteración

Simulador de un sistema de...

073 R: 5948 mejor: 4328 Comb: [11, 8, 5, 19, 7, 17, 2, 16, 13, 14, 0, 6, 1, 18, 4, 3, 9, 18, 15, 12] temp: 0.513

074 R: 6522 mejor: 4328 Comb: [11, 8, 10, 10, 17, 7, 16, 2, 13, 4, 0, 6, 1, 5, 14, 3, 9, 18, 15, 12] temp: 0.513

075 R: 6150 mejor: 4328 Comb: [19, 13, 10, 7, 8, 16, 11, 2, 17, 4, 0, 6, 1, 5, 14, 3, 9, 18, 15, 12] temp: 0.512

076 R: 5819 mejor: 4328 Comb: [19, 13, 10, 7, 8, 18, 15, 3, 6, 9, 1, 5, 17, 12, 14, 2, 4, 16, 11, 0] temp: 0.512

077 R: 5786 mejor: 4328 Comb: [19, 13, 10, 7, 8, 12, 0, 14, 5, 11, 6, 4, 3, 2, 16, 18, 1, 17, 9, 15] temp: 0.511

078 R: 6593 mejor: 4328 Comb: [0, 13, 10, 14, 18, 12, 19, 7, 3, 11, 6, 4, 5, 2, 16, 8, 1, 17, 9, 15] temp: 0.511

079 R: 5953 mejor: 4328 Comb: [0, 5, 10, 17, 18, 16, 19, 0, 3, 11, 6, 4, 13, 2, 12, 8, 1, 14, 7, 15] temp: 0.51

080 R: 5971 mejor: 4328 Comb: [0, 5, 10, 17, 18, 16, 19, 0, 3, 11, 6, 4, 13, 2, 12, 1, 8, 14, 7, 15] temp: 0.51

081 R: 6210 mejor: 4328 Comb: [0, 5, 10, 17, 18, 16, 19, 0, 3, 11, 6, 4, 13, 2, 1, 12, 8, 14, 7, 15] temp: 0.509

082 R: 6152 mejor: 4328 Comb: [0, 5, 10, 17, 18, 16, 19, 0, 3, 11, 6, 4, 2, 13, 1, 12, 8, 14, 7, 15] temp: 0.509

083 R: 5748 mejor: 4328 Comb: [0, 5, 10, 17, 18, 16, 19, 3, 9, 11, 6, 4, 2, 13, 1, 12, 8, 14, 7, 15] temp: 0.508

084 R: 5575 mejor: 4328 Comb: [0, 5, 10, 18, 17, 16, 19, 3, 9, 11, 6, 4, 2, 13, 1, 12, 8, 14, 7, 15] temp: 0.508

085 R: 6020 mejor: 4328 Comb: [0, 5, 10, 18, 17, 19, 16, 3, 9, 11, 6, 4, 2, 13, 1, 12, 8, 14, 7, 15] temp: 0.507

086 R: 5879 mejor: 4328 Comb: [0, 10, 5, 18, 17, 19, 16, 3, 9, 11, 6, 4, 2, 13, 1, 12, 8, 14, 7, 15] temp: 0.507

087 R: 5398 mejor: 4328 Comb: [0, 10, 5, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 1, 12, 8, 14, 7, 15] temp: 0.506

088 R: 5927 mejor: 4328 Comb: [10, 0, 5, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 1, 12, 8, 14, 7, 15] temp: 0.506

089 R: 6126 mejor: 4328 Comb: [10, 0, 5, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 1, 8, 12, 14, 7, 15] temp: 0.505

090 R: 6101 mejor: 4328 Comb: [10, 5, 0, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 1, 8, 12, 14, 7, 15] temp: 0.505

091 R: 6126 mejor: 4328 Comb: [10, 0, 5, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 1, 8, 12, 14, 7, 15] temp: 0.504

092 R: 6107 mejor: 4328 Comb: [10, 0, 5, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 8, 1, 12, 14, 7, 15] temp: 0.504

093 R: 6082 mejor: 4328 Comb: [10, 5, 0, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 8, 1, 12, 14, 7, 15] temp: 0.503

094 R: 6216 mejor: 4328 Comb: [10, 5, 0, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 8, 1, 14, 12, 7, 15] temp: 0.503

095 R: 5906 mejor: 4328 Comb: [5, 10, 0, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 8, 1, 14, 12, 7, 15] temp: 0.502

096 R: 5834 mejor: 4328 Comb: [5, 10, 0, 18, 17, 19, 16, 3, 11, 9, 6, 4, 2, 13, 8, 1, 14, 7, 12, 15] temp: 0.502

097 R: 6315 mejor: 4328 Comb: [5, 10, 0, 18, 17, 19, 16, 3, 9, 11, 6, 4, 2, 13, 8, 1, 14, 7, 12, 15] temp: 0.501

098 R: 6180 mejor: 4328 Comb: [5, 10, 0, 18, 17, 19, 16, 3, 9, 11, 6, 4, 2, 8, 13, 1, 14, 7, 12, 15] temp: 0.501

099 R: 6301 mejor: 4328 Comb: [5, 10, 0, 18, 17, 19, 16, 3, 9, 11, 6, 4, 2, 8, 13, 14, 1, 7, 12, 15] temp: 0.5

La mejor combinación es: [5, 10, 0, 18, 17, 19, 16, 3, 9, 11, 6, 4, 2, 8, 13, 14, 1, 7, 12, 15] y su puntaje es 4328

Confidencialidad

Editor

Reutilizar archivos

Confidencialidad

Editor

Reutilizar archivos

Historial

Memoria

44351 + 10 =

4,435.1

4454 + 4520 + 4434 + 3751 +

4554 + 4442 + 4476 + 4726 +

4666 + 4328 =

44,351

Para el primer problema discreto obtiene un resultado promedio de: 4435.1 con 10 iteraciones