

PRÁCTICA 10

Algoritmos genéticos

Grupo 9:

Luis Alfonso González de la Calzada

Alfredo Hernández Burgos

Sobre la población inicial en los algoritmos genéticos y el código de la práctica:

En primer lugar hay que explicar que para los algoritmos genéticos usamos una población inicial especial de 38 círculos. Corresponden a los círculos más grandes que entran en las cuadrículas $2 \times 2 = 4$ círculos, $3 \times 3 = 9$ círculos y $5 \times 5 = 25$ círculos. Hacemos esto para tener una población inicial diversa y que así los cruces sean más interesantes. Si el usuario quiere puede usar una configuración de círculos completamente aleatoria como población inicial, pero si selecciona “0” se empezará con esta.

El código de la práctica está comentado exhaustivamente y todas las explicaciones respectivas a como está hecho se encuentran allí.

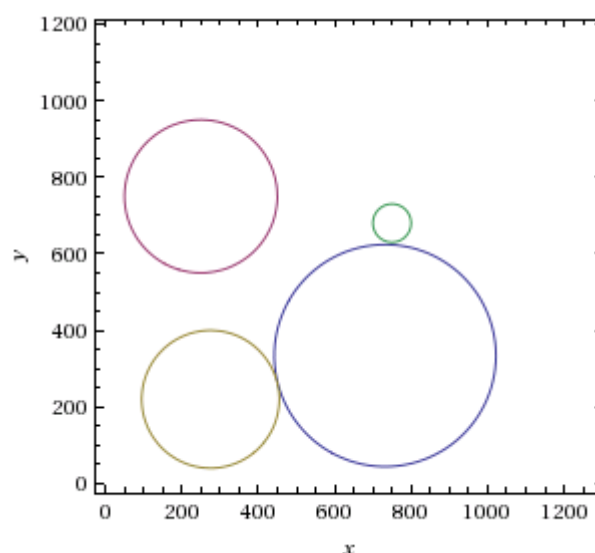
Problema 1: Algoritmos genéticos:

El problema 1 está definido en “inicial.txt”. Después de muchas pruebas jugando con las tasas de cruce y de mutación, el porcentaje de población de élite y el corte a nivel de gen o de individuo hemos llegado a la conclusión de que guardar una población de élite entre el 5% y el 10% del total y usar una tasa de mutación entre el 0,3 y el 0,5 da los mejores resultados. La tasa de cruce funciona bien a partir de 0,6 y el corte debe ser a nivel de gen.

La tasa de cruce a 0,7 permite que los cruces ocurran con suficiente frecuencia para dar posibilidad a reproducirse a individuos con menos fitness (la ruleta ocurre más veces). Es crucial guardar una población de élite en equilibrio con una tasa de mutación adecuada. Parece que 0,3 de tasa de mutación y 5% de población de élite funciona bien. Una tasa de mutación menor produce mínimos locales en el radio de los círculos y una población de élite mayor del 10% afecta al cruce. Cuando se produce el corte hemos decidido que debe ser a nivel de gen ya que a nivel de individuo (cortar entre posX y posY o hacerlo entre posY y radio) hace que la población se diversifique de manera mucho más lenta.

La configuración del problema y los resultados aparecen a continuación. El círculo morado es el que representa la solución (también lo es en el resto de gráficos), que es prácticamente perfecta, quedándose sólo a 1 píxel del radio óptimo.

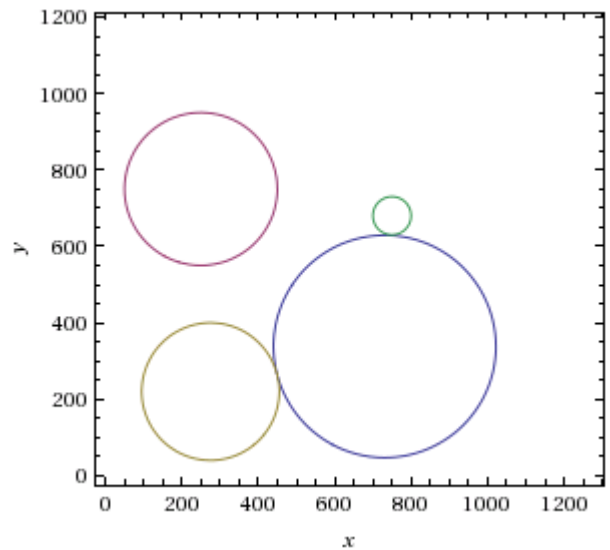
```
+-----+
| RESUMEN ALGORITMOS GENÉTICOS PROBLEMA 1 |
+-----+
| Configuración:                             |
+-----+
| -Individuos por generación: 38             |
| -Generaciones: 10000                      |
| -Tasa de cruce: 0.7                       |
| -Tasa de mutación: 0.3                    |
| -Porcentaje élite: 5%                     |
| -Corte a nivel gen                         |
+-----+
| Resultados:                               |
+-----+
| -Círculo solución: [x=732, y=334, r=290]  |
+-----+
```



Problema 1: Fuerza bruta:

Por fuerza bruta obtenemos obviamente una solución perfecta. Trabajar en un plano de 1024x1024 píxeles lo hace aceptable. Sin embargo en planos mayores el tiempo de computación crecería exponencialmente. Sería ahí donde los algoritmos genéticos ganarían la partida.

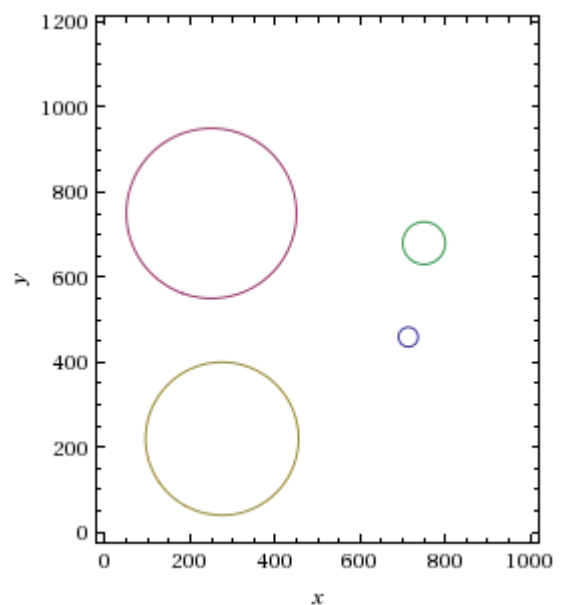
```
+-----+
| RESUMEN FUERZA BRUTA PROBLEMA 1 |
+-----+
| Círculos generados: 536870912 |
| Círculos solución: 292 |
| Círculos no solución: 536870620 |
+-----+
| Resultados: |
+-----+
| Solución final: [x=731, y=338, r=291] |
+-----+
```



Problema 1: Aleatorio:

Generando 25 círculos aleatoriamente conseguimos que 2 de ellos sean solución. Sin embargo la mejor de ellas es una solución muy pobre. Realmente este método es cuestión de suerte, mientras más pruebas más posibilidades. No hay mucho más que decir.

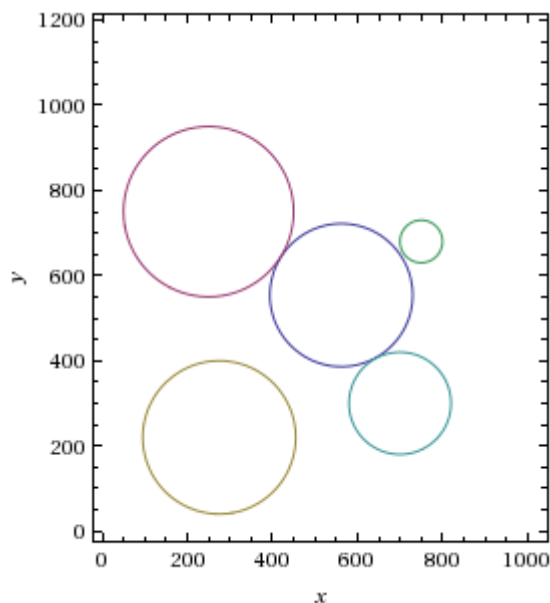
```
+-----+
| RESUMEN ALEATORIO PROBLEMA 1 |
+-----+
| Círculos generados: 25 |
| Círculos solución: 2 |
| Círculos no solución: 23 |
+-----+
| Resultados: |
+-----+
| Solución final: [x=713, y=459, r=23] |
+-----+
```



Problema 2: Algoritmos genéticos:

El problema 2 está definido en “inicial2.txt”. Este problema añade un nuevo círculo al problema 1 haciendo que las soluciones sean completamente distintas. La configuración del problema y los resultados aparecen a continuación. Como vemos en el gráfico la solución parece ser óptima. Pero sólo lo parece. Es una solución óptima sólo en esa zona y lo veremos en el resultado obtenido por fuerza bruta.

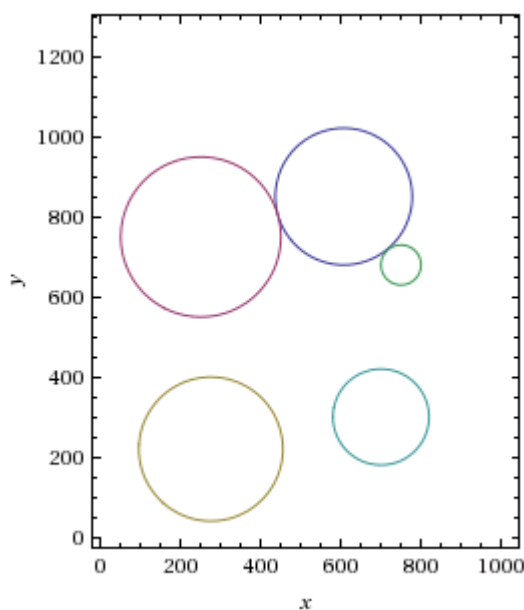
```
+-----+
| RESUMEN ALGORITMOS GENÉTICOS PROBLEMA 2 |
+-----+
| Configuración:                          |
+-----+
| -Individuos por generación: 38          |
| -Generaciones: 10000                   |
| -Tasa de cruce: 0.7                    |
| -Tasa de mutación: 0.3                 |
| -Porcentaje élite: 5%                  |
| -Corte a nivel gen                     |
+-----+
| Resultados:                            |
+-----+
| -Círculo solución: [x=562, y=554, r=168]
+-----+
```



Problema 2: Fuerza bruta:

Es curioso que creyendo haber encontrado la solución perfecta la fuerza bruta nos enseñe que estábamos equivocados. La nueva solución está en una parte completamente distinta a la obtenida mediante algoritmos genéticos y sólo tiene 3 píxeles más de radio que el círculo obtenido anteriormente. Es un claro ejemplo de que cuando un algoritmo genético encuentra una solución local muy buena le va a costar encontrar la óptima (probablemente necesitaría una mutación muy oportuna).

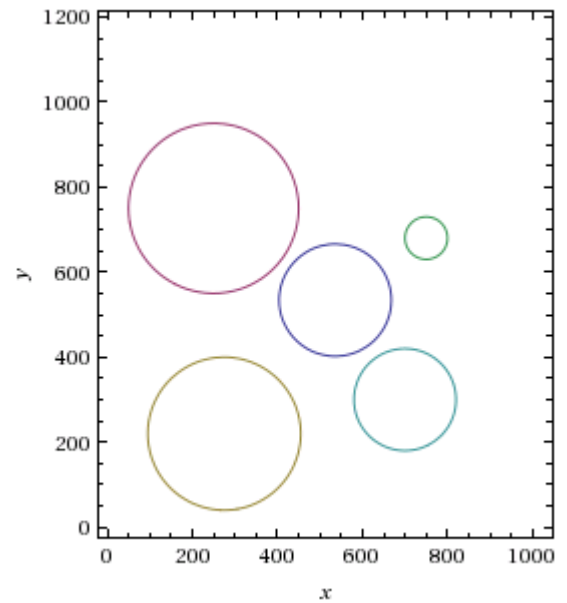
```
+-----+
| RESUMEN FUERZA BRUTA PROBLEMA 2        |
+-----+
| Círculos generados: 536870912          |
| Círculos solución: 172                 |
| Círculos no solución: 536870740       |
+-----+
| Resultados:                            |
+-----+
| Solución final: [x=607, y=851, r=171]
+-----+
```



Problema 2: Aleatorio:

Generando 25 círculos aleatoriamente conseguimos que 1 de ellos sea solución, y de hecho una bastante aceptable. Como hemos dicho antes no es más que una cuestión de suerte. Antes no la tuvimos y la solución era mala y ahora sí la hemos tenido y la solución es aceptable.

```
+-----+
| RESUMEN ALEATORIO PROBLEMA 2 |
+-----+
| Círculos generados: 25      |
| Círculos solución: 1       |
| Círculos no solución: 24   |
+-----+
| Resultados:                 |
+-----+
| Solución final: [x=536, y=534, r=132] |
+-----+
```



Sobre las gráficas:

Se han conseguido mediante el uso de “plot” en <http://www.wolframalpha.com/>. Los comandos de ploteo aparecen al ejecutar el código al final de cada resumen del algoritmo utilizado.