

# Inteligencia Artificial [2022S1]



[Área personal](#) / [Mis cursos](#) / [IA2022S1](#) / [III. Optimización Inteligente](#) / [Proyecto 3: Optimizacion](#)

## Proyecto 3: Optimizacion

**Apertura:** miércoles, 20 de abril de 2022, 00:00

**Cierre:** martes, 17 de mayo de 2022, 00:00

Marcar como hecha

En este proyecto se requiere programar uno de los siguiente algoritmos de optimización para resolver el problema de asociado (revisar la puntuación asociada):

Algoritmo	Problema	Porcentaje
Univariate Marginal Distribution Algorithm	The Satisfiability Problem	70%
Genetic Algorithm (Binary Codification)	The Satisfiability Problem	80%
Continuous-Gaussian Univariate Marginal Distribution Algorithm	Circle Detection Problem	90%
Genetic Algorithm (Binary Encoding - Integer Decoding)	Circle Detection Problem	100%

### Pasos a seguir para la solución de los problemas

#### *The Satisfiability Problem*

El algoritmo será aplicado a resolver el problema de satisfacción; será trabajado como un problema de maximización. El problema está basado en lógica proposicional, el contexto del problema puede ser revisado en el siguiente [LINK](#). Los pasos a seguir son:

- Cargar cualquier instancia de la sección *Uniform Random 3-SAT* de la página del [SATLIB-Benchmark Problems](#). Sugerencia: cargar el problema en una matriz con 3 columnas y la cantidad de renglones como clausulas marque la instancia, por ejemplo; uf20-91, indica que serán 91 clausulas de 3 variables cada una; el tamaño de cada solución candidata es de 20 (la cantidad de variables del problema). En cada celda indicar el índice de la variable y un signo menos si es una negación para poder asociar adecuadamente el problema.
- La función objetivo manejarla como la sumatoria de las clausulas verdaderas que se obtienen con la configuración de una solución candidata, en lugar de solo asignar un 1 o un 0.
- Reportar la mejor solución obtenida y su valor aptitud.

#### *Circle Detection Problem*

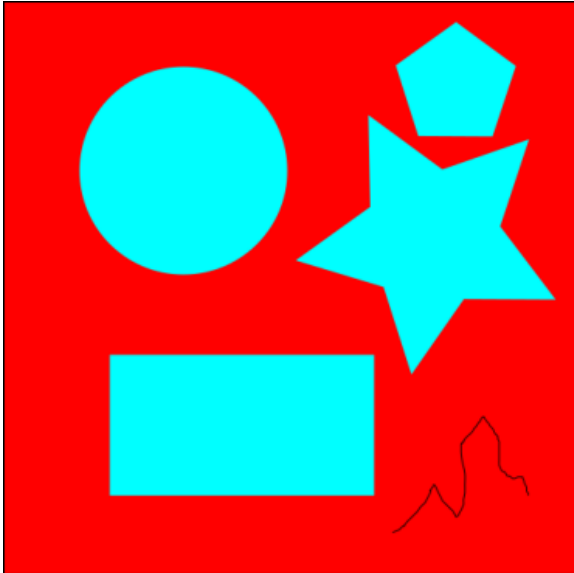
El algoritmo será aplicado a la detección de un círculo en imágenes; será trabajado como un problema de minimización. Los pasos a seguir (con base a las referencias en el archivo CircleDetection.zip) son:

- Cargar cualquier imagen (revisar las imágenes proporcionados en el archivo Circle Detection.zip); en opencv la función es `imread(ruta)`.
- Obtener los canales de la imagen ( $R, G, B$ ); en opencv la función es `split(img)` (regresa los canales como  $B, G, R$ , con la

- función *merge([R,G,B])* se puede generar la imagen nuevamente con los canales como se indica en el argumento.
- Preprocesamiento de imagen (se puede usar librería como opencv):
    - Convertir la imagen a escala de grises, usando  $g = 0.299R + 0.587G + 0.114B$ ; cada canal debe tener el valor calculado para  $g$ .
    - Quitar ruido mediante filtro suavizado de mediana con ventana  $N = 3$ ; en opencv la función es *medianBlur(img, 3)*.
    - Detectar bordes mediante el algoritmo de Canny, con ambos parámetros con valor de 50; la función en opencv es *canny(img, 50,50)*.
  - Generación del espacio de búsqueda a partir de la imagen
    - Obtener contornos de imagen, es una lista de los bordes continuos, si se detectan 3 conjuntos de pixeles (siluetas u objetos) continuos, cada conjunto contiene las coordenadas  $(x, y)$  de los bordes que conforman el contorno; en opencv la función es *findContours(img,RETR\_EXTERNAL,CHAIN\_APPROX\_SIMPLE)*.
    - Generar una lista de pixeles bordes la cual contenga las coordenadas de todos los pixeles de los contornos encontrados (sin estar agrupada por un contorno); esto forma el espacio de búsqueda del tamaño como cantidad de pixeles bordes existan en esta lista.
  - La configuración para las soluciones candidatas de la metaheurística es: cada solución debe poder codificar 3 variables enteras (ajustar para el GA o usar valores reales y redondear para CG-UMDA), cada uno de estas debe poder tener un valor entre 0 y la cantidad de pixeles bordes menos uno.
  - Función objetivo:
    - Asignar cada una de las variables de la solución a las variables  $i, j$  y  $k$ ; estas representan los índices de pixeles borde en la lista de coordenadas
    - Comprobar que  $i \neq j \neq k$ , si no cumple asignar la aptitud de esa solución como 1
    - Calcular el centro y radio de acuerdo a las coordenadas de los pixeles borde en los índices  $i, j$  y  $k$ . Usar ecuaciones 1 – 3 de la referencia *Gonzalez2021*. Si el denominador usado para calcular las componentes del centro devuelve 0 o el radio es 0, terminar evaluación y asignar aptitud de esa solución como 1. De lo contrario, continuar evaluación.
    - Generar y almacenar en  $tp$  los puntos de prueba a partir del algoritmo Mid-point Circle Algorithm (MCA):
      - Entradas  $x0, y0, r$  (componentes del centro del círculo y radio; respectivamente)
      - Hacer  $x = r, y = 0$
      - Almacenar el primer punto  $(x + x0, y + y0)$  y reflejar en los demás octantes  
 $\{(-x + x0, y + y0), (-x + x0, -y + y0), (x + x0, -y + y0), (y + x0, x + y0), (-y + x0, x + y0), (-y + x0, -x + y0)\}$   
 y almacenar (puntos con desplazamiento)
      - Hacer  $P = 1 - r$
      - Mientras  $x > y$  hacer:
        - Hacer  $y = y + 1$
        - Si  $P \leq 0$  (punto en o dentro del perímetro del círculo) hacer  $P = P + 2 * y + 1$  sino hacer  $x = x - 1$  y  $P = P + 2 * y - 2 * x + 1$
        - Si  $x < y$  entonces terminar ciclo
        - Almacenar puntos  $\{(x + x0, y + y0), (-x + x0, y + y0), (-x + x0, -y + y0), (x + x0, -y + y0)\}$
        - Si  $x \neq y$  entonces almacenar puntos  
 $\{(y + x0, x + y0), (-y + x0, x + y0), (-y + x0, -x + y0), (y + x0, -x + y0)\}$
      - Salida lista de puntos generados (puntos prueba)
    - Obtener el tamaño de  $tp$  en  $npe$
    - Si  $npe = 0$  entonces terminar evaluación y asignar aptitud de esa solución como 1
    - Crear lista  $ev$  y almacenar en ella la cantidad de coordenadas de puntos de prueba almacenados en  $tp$  que coinciden con los pixeles borde de cada contorno obtenido en la imagen; es decir, si la imagen tiene 3 contornos,  $ev$  debe almacenar 3 cantidades.
    - Obtener el máximo valor contenido en  $ev$ , negar su valor y dividirlo entre  $npe$ . Asignar este valor como aptitud de la solución actual. Esta función objetivo da prioridad al contorno al cual mejor se parece nuestra solución (en teoría, un círculo).
  - Reportar resultados. Mostrar la imagen original con el círculo detectado; en opencv la función es *circle(img,(x0,y0),r,(0,0,255), 5)*. Mostrar el centro y radio obtenido y la aptitud lograda de la mejor solución candidata.

A continuación se muestran algunos resultados del proceso de detección de círculos

- Imagen Original (c7.png)



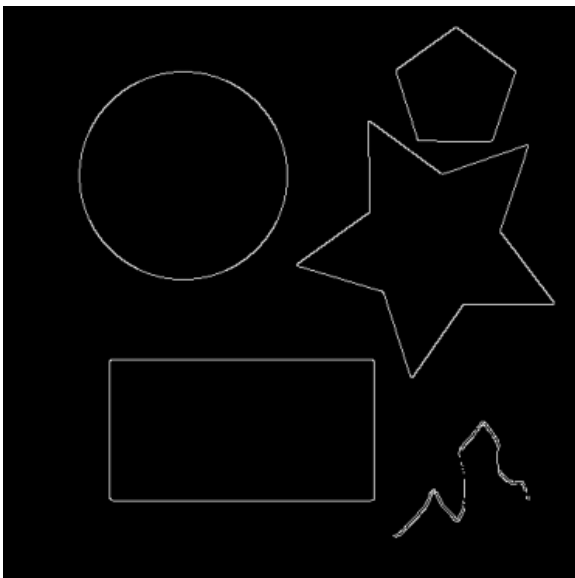
- Imagen en escala de grises



- Imagen con reducción de ruido (suavizado)



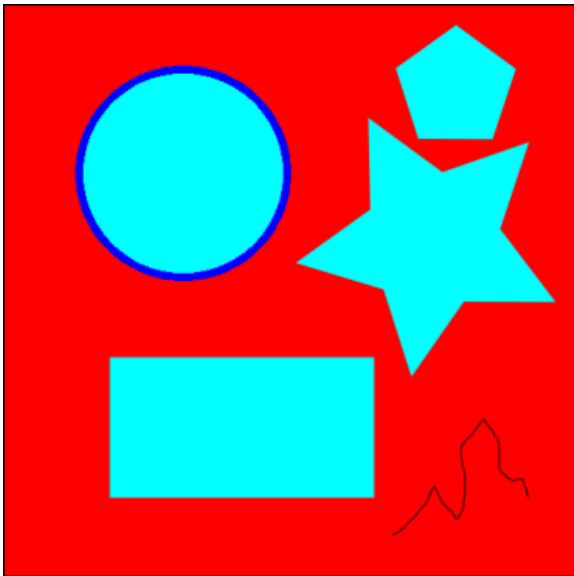
- Imagen con detección de bordes



- Imagen con identificación de contornos (en color verde)



- Imagen con circulo detectado (en color azul)



Coordenadas reportadas para el centro y radio del circulo:

x	y	r
159.627329	150.931677	92.0140793

Coordenadas encontradas por el proceso de optimización:

```
Best solution found:  
X = [823.58531744 652.52401846 810.37677065]  
F = [-0.35384615]  
c= (159,151), r = 92
```

Envío: Debe enviar su proyecto en una carpeta comprimida zip con el nombre formado de lo siguiente:  
IA\_3Optimizacion\_ApellidoPaterno-ApellidoMaterno (por ejemplo: IA\_3Optimizacion\_Espinal-Jimenez)

**NOTA #1:** Investigar librerías para generación de números aleatorios, manipulación de archivos e imágenes. Lo esencial del algoritmo debe ser autoría del estudiante.

**NOTA #2:** Para su revisión el código debe correr sin errores, de lo contrario la calificación será de cero en este proyecto.

**NOTA #2:** Su entrega debe realizar todos los pasos definidos de manera adecuada para acumular el total asignado al algoritmo seleccionado. De no cumplir será evaluado con cero.

 [Circle Detection.zip](#)

19 de abril de 2022, 20:11

## Estado de la entrega

**Estado de la entrega** No entregado

**Estado de la calificación** Sin calificar

**Tiempo restante** 16 horas 13 minutos

**Última modificación** -

**Comentarios de la entrega** ▶ [Comentarios \(0\)](#)

Agregar entrega

Todavía no has realizado una entrega.

◀ Proyecto 2: Agrupamiento

Ir a...

Usted se ha identificado como Pedro Antonio Vázquez (Cerrar sesión)

Reiniciar tour para usuario en esta página

IA2022S1

Resumen de retención de datos

Descargar la app para dispositivos móviles