

# Inteligencia Artificial 1

## Proyecto Final

### BOXOPT

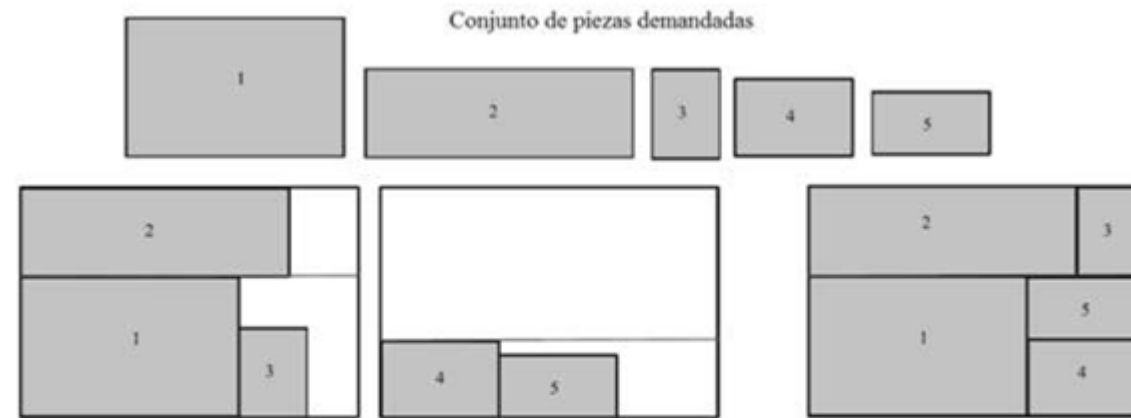
- Ordenamiento de cajas en contenedores
  - Diego Rinaldo Cazon Condori
- Facultad de Ingeniería – Universidad Nacional de Cuyo - 2022

# Índice

- 1. Definición del problema
- 2. Métodos usados
  - 2.1 Árbol de ordenamiento y Algoritmo de Ordenamiento
  - 2.2 Árbol de corte guillotina
  - 2.3 Algoritmo Genético
- 3. Diseño Experimental
- 4. Resultados
- 5. Conclusiones
- 6. Bibliografía

# 1. Definición del problema

- El problema a resolver es el de ordenamiento de cajas en contenedores pero en un espacio bidimensional, similar al ordenar placas en una placa mayor



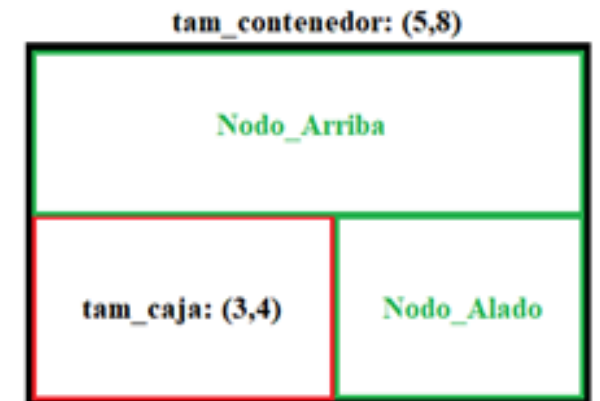
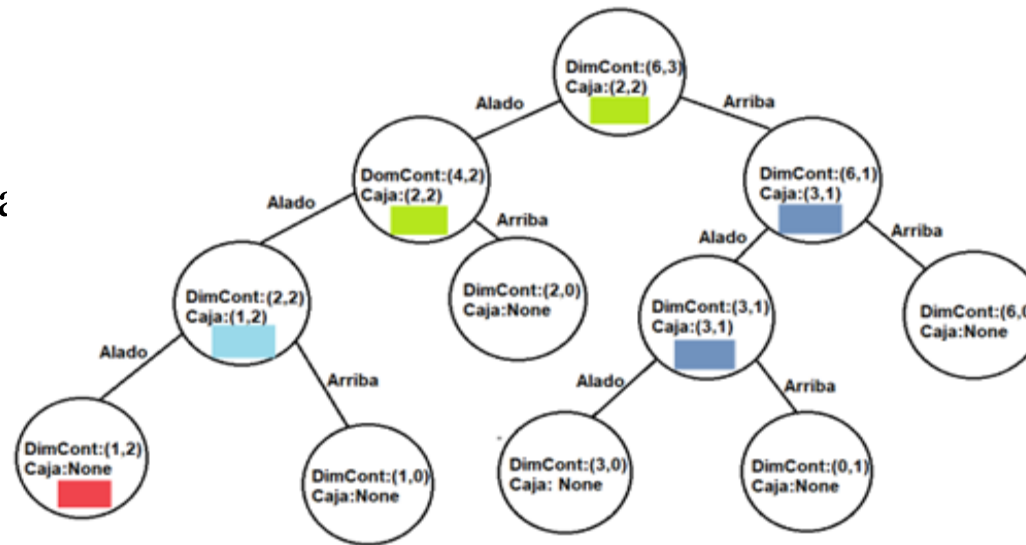
- Estos problema de ordenamiento, entran dentro de la familia de los problemas de corte y empaquetamiento bidimensional en una sola placa (2D-CSP).
- Uno de los objetivos a cumplir, es el de disminuir el espacio de área desperdiciado y además mantener un tiempo de ejecución relativamente bajo.

## 2. Metodos Usados

## 2.1 Árbol de ordenamiento

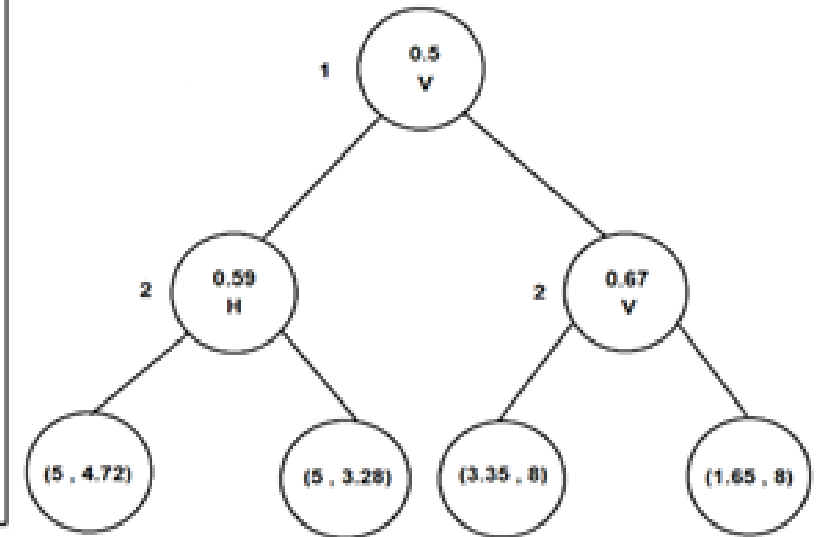
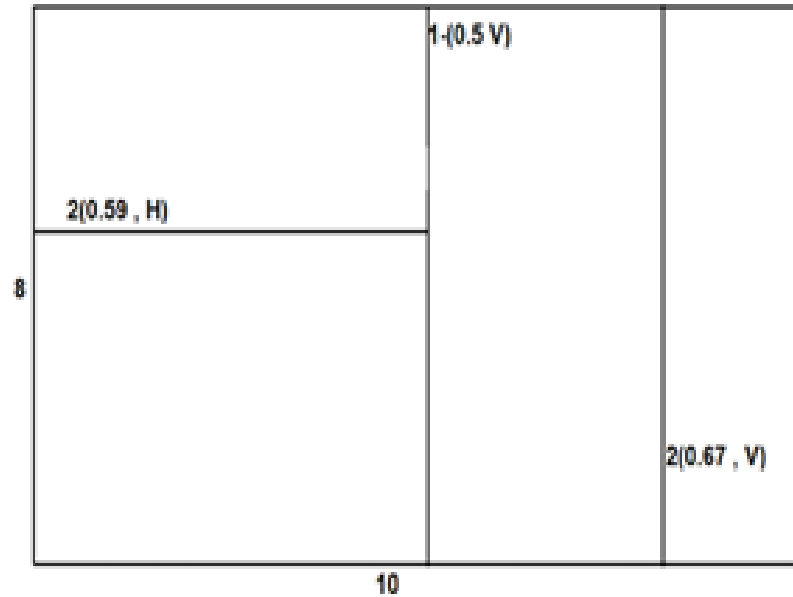
- El árbol de ordenamiento es un árbol binario el cual tiene como objetivo dar información de cómo se encuentran ordenadas las cajas, espacios aun no recorridos y espacios desperdiciados

- Atributos de Nodo:
  - Tam\_Contenedor:
  - Tam\_Caja\_Ingresada:
  - Nodo\_Alado
  - Nodo\_Arriba



## 2.2 Árbol de corte guillotina

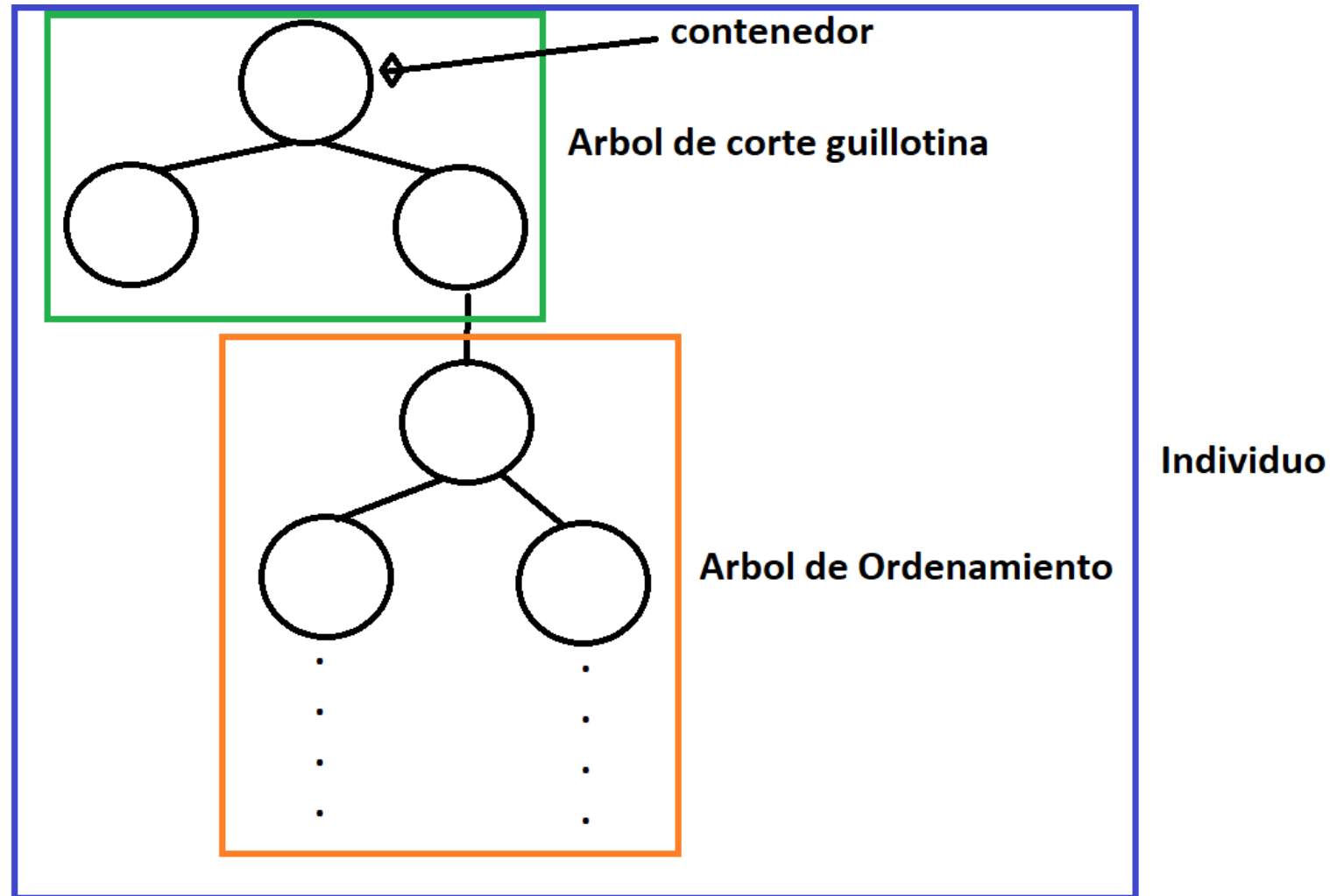
- El árbol de Corte Guillotina es un árbol binario completo, esta estructura nos permitirá dividir el contenedor principal en varios subcontenedores.
- Usaremos cada subcontenedor para ordenar las cajas en estos, tratando de obtener esos subcontenedores para un conjunto de cajas que dejen menos espacios desperdiciados.
- La altura del árbol define la cantidad de cortes que se realizarán sobre el contenedor
- Atributos de Nodos:
  - Tipo\_de\_Corte
  - Porcentaje\_de\_Corte
  - Nodo\_Derecho
  - Nodo\_Izquierdo



## 2.3 Algoritmo Genético

- Llamados así porque se inspiran en la evolución biológica
- **Inicialización**
- **Condición de término**
- **Evaluación**
- **Selección**
- **Cruzamiento**
- **Mutación**
- **Reemplazo**

# Individuo de la población



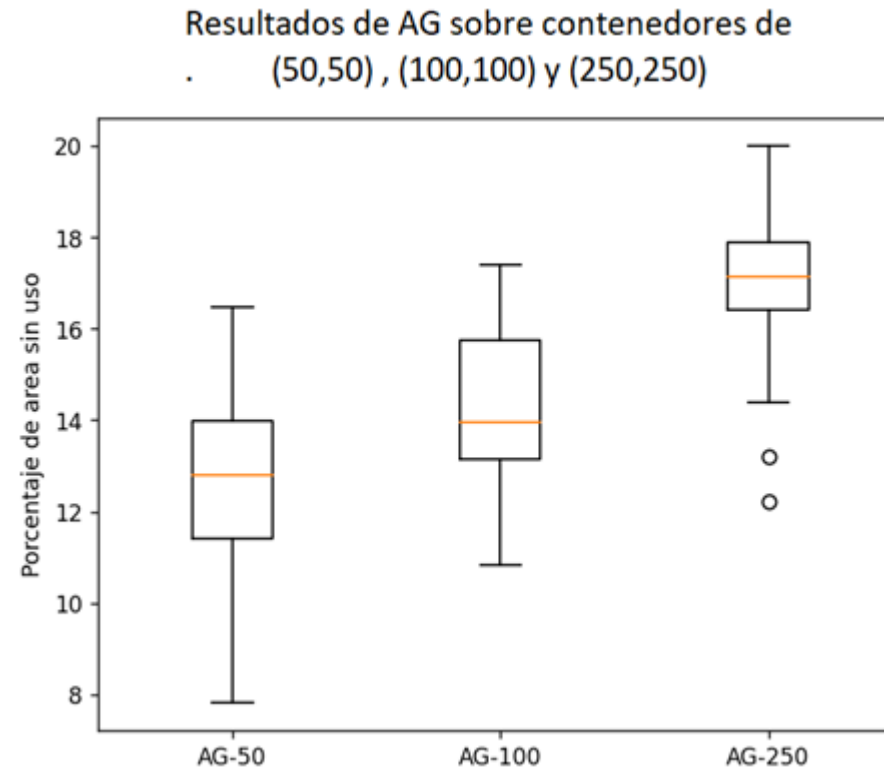


### 3. Diseño Experimental

- La prueba consiste en varios escenarios y una competencia entre el algoritmo propuesto y un algoritmo que solo realizara búsquedas aleatorias.
- Definimos 3 escenarios, un escenario con contenedor (50 x 50), uno de (100 x 100) y un escenario con un contenedor de (250 x 250), con 30 conjuntos de cajas a ordenar,
- Para las pruebas que realizaremos proponemos los siguientes parámetros , altura de los arboles guillotina 3, Cantidad de Individuos 125 y Cantidad de ciclos 50.
- El algoritmo adversario con el que pondremos a competir al algoritmo propuesto, es uno que no cuenta con el AG, sino que en su lugar contara con una búsqueda aleatoria.

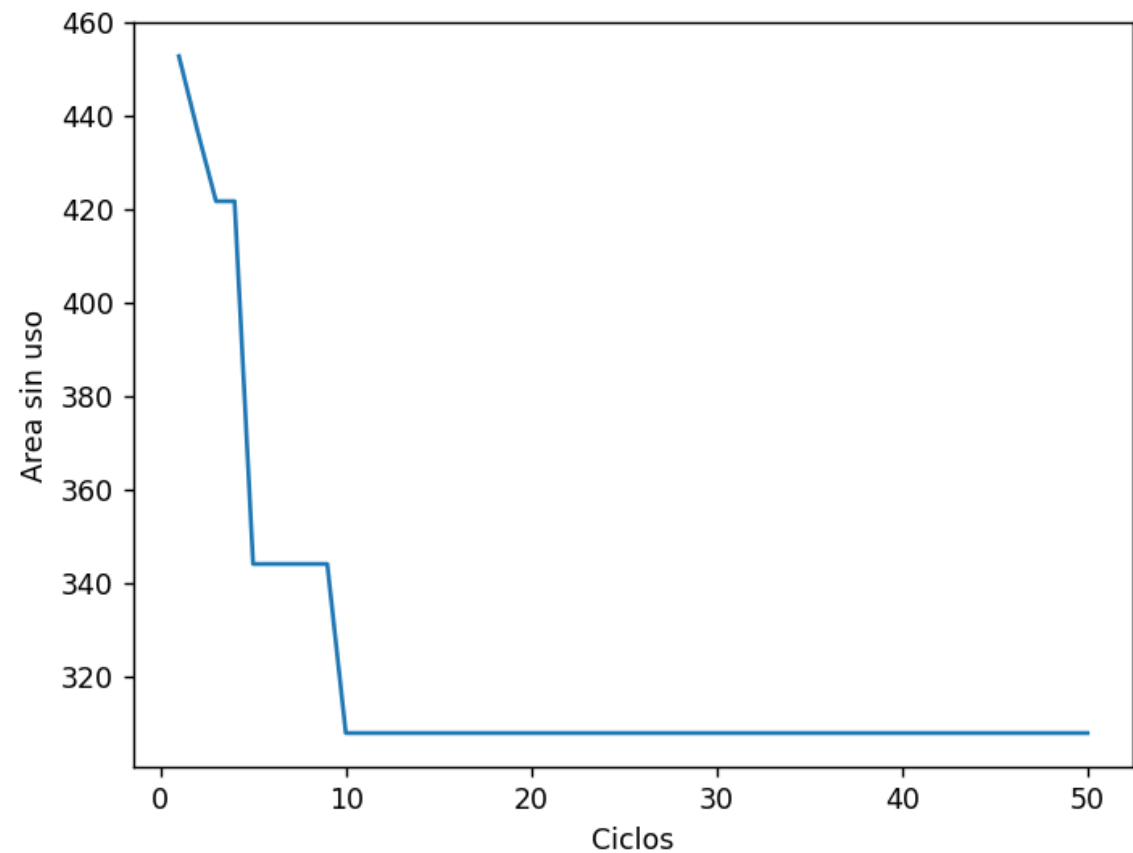
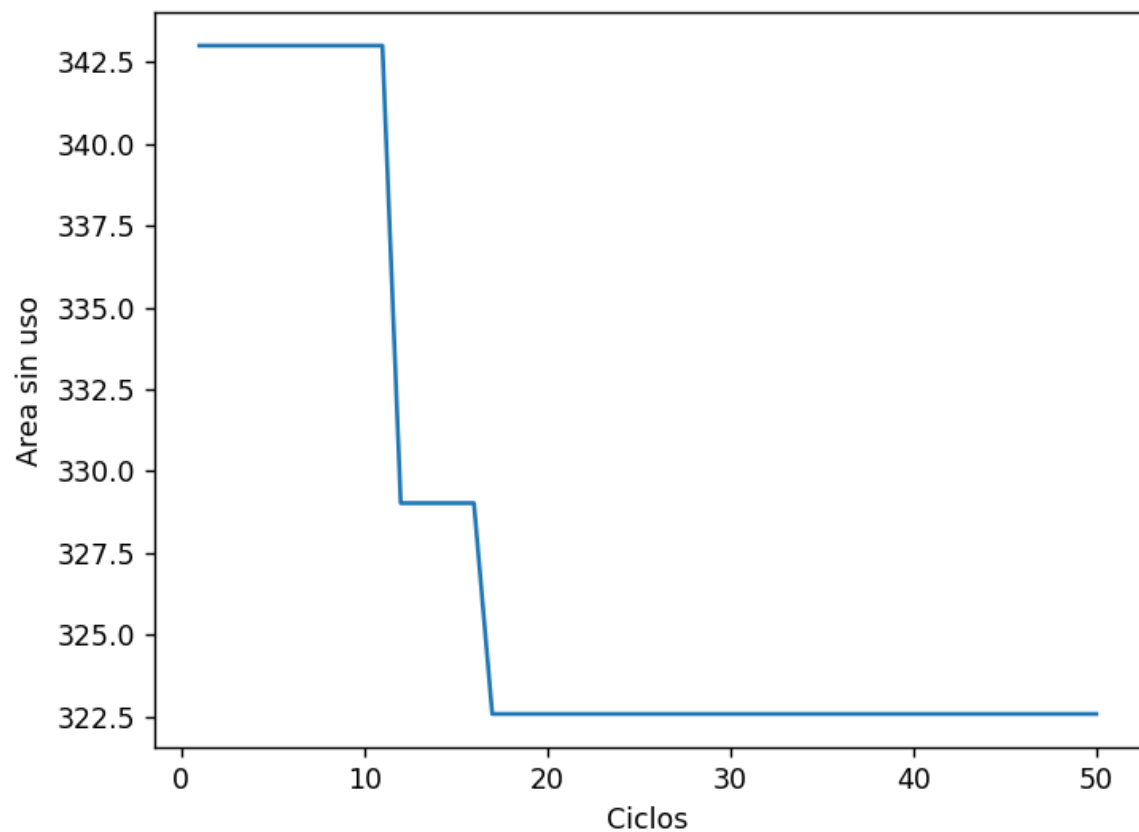
## 4. Resultados Obtenidos

- Ahora mostraremos los resultados obtenidos

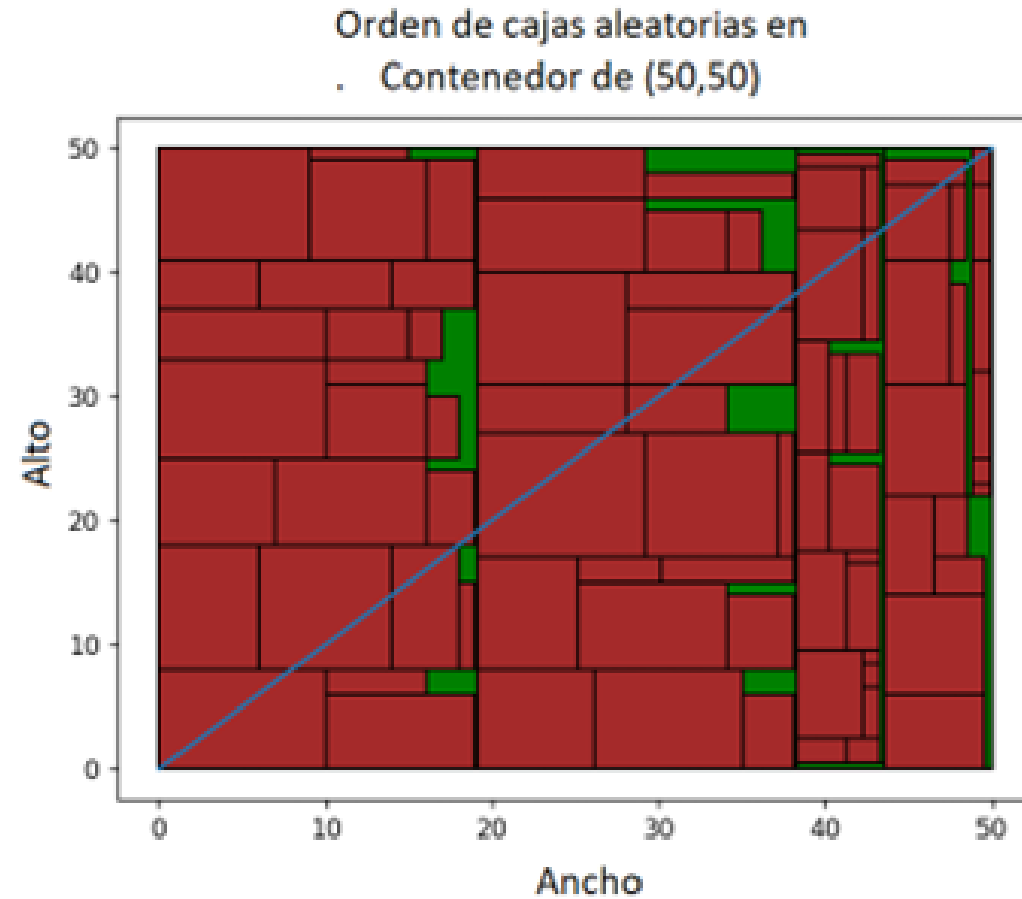


- Con los resultados obtenidos vemos que mientras mas aumenta el tamaño de los contenedores obtenemos peores resultados.

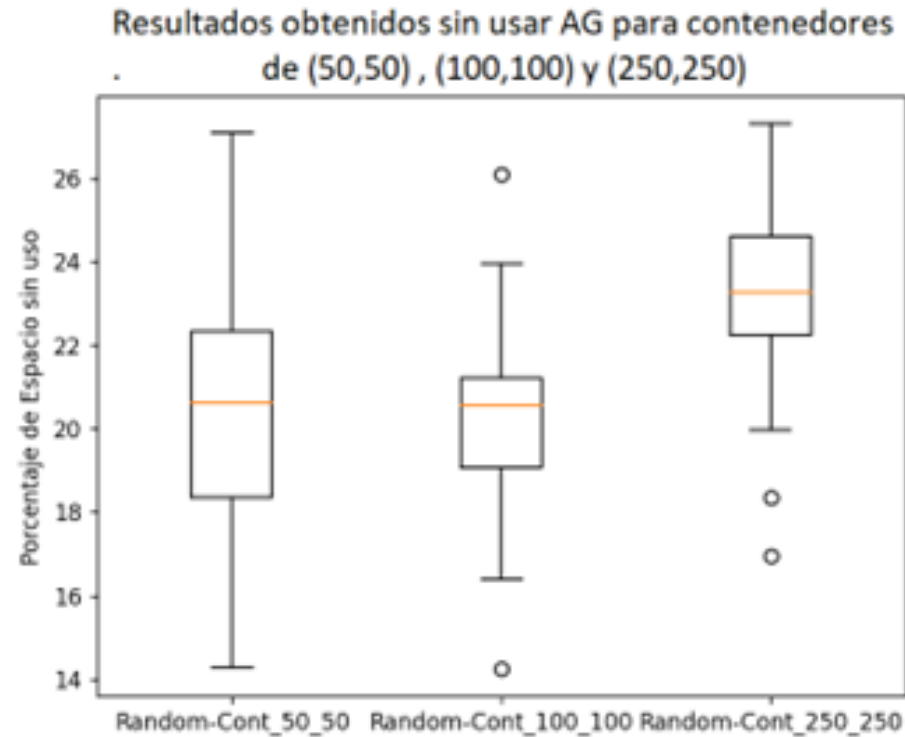
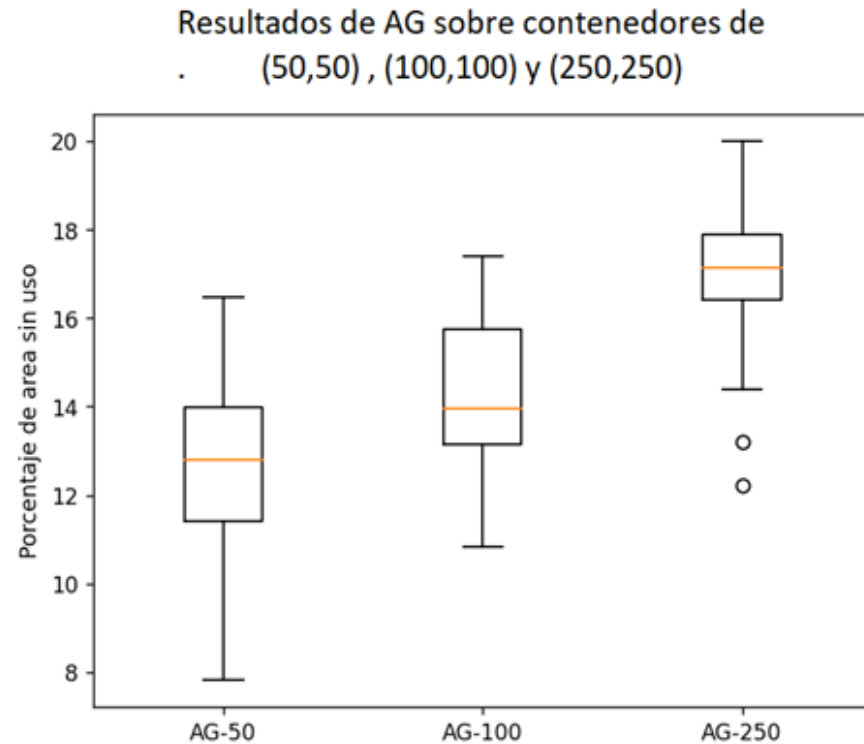
Ahora mostraremos algunos resultados de como se reduce el area sin uso, con el pasar de los ciclos para un contenedor de (50x50)



La imagen representa el orden en final que quedaria un contenedor luego de usar el algoritmo, los espacios verdes son los espacios desperdiciados y las cajas marrones, las cajas a ordenar.



- Ahora haremos la comparación del algoritmo propuesto con AG y un algoritmo que busca aleatoriamente, para los mismos escenarios de 50, 100 y 250.



Los resultados obtenidos, para el algoritmo aleatorio, suelen ser muy dispersos, en comparación con el propuesto además de búsquedas aleatorias al inicio, agrega una búsqueda mas profunda a medida que la poblacion avoluciona, en espacios de soluciones cada vez mas reducidos.

# 5. Conclusiones

- El algoritmo propuesto que hace uso de un algoritmo genético, tiene mejor desempeño que sin usar AG, los resultados mejoran en alrededor del 10%.
- Podemos concluir que el uso del AG es importante para mejorar los resultados ya que este en un comienzo realiza una búsqueda aleatoria y luego con la evolución de la población explora mejor los espacios de soluciones
- El algoritmo propuesto nos da buenos resultados, a pesar de ser bastante simple, sabemos que estos resultados pueden mejorar aún más:
  - Aumentando la cantidad de individuos de la población de AG.
  - Implementando programación paralela para reducir los tiempos,
  - Implementando PSO para mejorar los porcentajes de corte,
  - Programando a cada individuo un algoritmo de búsqueda local que busque sus mejores porcentajes de corte en cada ciclo.

# 6. Bibliografia

- 1- PROBLEMA DE EMPAQUETAMIENTO RECTANGULAR BIDIMENSIONAL TIPO GUILLOTINA - Universidad Tecnológica de Pereira
- 2- SOLUCIÓN DEL PROBLEMA DE EMPAQUETAMIENTO ÓPTIMO BIDIMENSIONAL EN UNA SOLA PLACA – David Alvarez Martinez
- 3- EL CODIGO DEL PROYECTO LO PUEDE ENCONTRAR EN [https://github.com/DiegoCC14/DiegoCazon\\_ProyectoFinal\\_IA1](https://github.com/DiegoCC14/DiegoCazon_ProyectoFinal_IA1)