

# Metaheurísticas - Proyecto Final

## *Battle Royale Metaheuristic*

### *Cec17 Competition*

Sergio Quijano Rey - 72103503k  
sergioquijano@correo.ugr.es  
4º Doble Grado Ingeniería Informática y Matemáticas

June 25, 2021

# Contenidos

## 1 Identificación del problema a resolver

- Software de Daniel Molina
- Casos de uso a estudiar

## 2 Nuestra metaheurística

- Código de la metaheurística
- Inspiración de la metaheurística
- Dinámicas del juego
- Decisiones basadas en la ingeniería

# Problema a resolver

El problema a resolver consiste en realizar una propuesta de metaheurística original para problemas de codificación real. A partir de esta propuesta, realizaremos una implementación de dicha metaheurística original, y trabajaremos la competición *Cec17*.

# Software de Daniel Molina

- Para trabajar con la competición *Cec17*
- Se encuentra en <https://github.com/dmolina/cec2017real>

El software contiene:

- Librería escrita en C que define las 31 funciones de *fitness* que debemos optimizar, para las distintas dimensiones disponibles, así como otras funcionalidades
- Código python para generar unas tablas *Excel* que podemos usar en [tacolab.org](http://tacolab.org) para comparar con otros algoritmos de referencia
- Wrapper para Python, con el que podemos acceder a todas las funciones definidas en la ya mencionada librería dinámica

# Casos de uso a estudiar

Por no tener que lidiar con tiempos de ejecución muy largos, los profesores de prácticas nos indican:

- Debemos estudiar el comportamiento sobre 31 funciones distintas
- Debemos trabajar con dimensión 10 y 30
- Por cada función y dimensión, debemos lanzar 10 ejecuciones de la búsqueda

# Nuestro código

- Nuestro código se encuentra alojado en un repositorio de *Github*
- <https://github.com/SergioQuijanoRey/PracticaFinalMetaheurísticas>

# Inspiración de la metaheurística

- Basado en el género de videojuegos *Battle Royale*
- Este género de videojuegos se inspira a su vez en la trilogía *Los juegos del hambre*





# Dinámicas del juego

Las dinámicas que introducen estos juegos son:

- 50 o 100 jugadores compitiendo
- Los jugadores aparecen en el mapa
- Fase inicial de recolección de recursos
- Fase final en la que el círculo se cierra
- Batallas entre dos jugadores
- Jugadores que reviven aleatoriamente

## 50 o 100 jugadores compitiendo

- Una gran cantidad de jugadores compiten por ser los últimos vivos en la partida
- Por tanto, nuestra metaheurística va a tener un carácter poblacional

# Los jugadores aparecen en el mapa

- Modelamos esta primera fase como un fenómeno aleatorio
- Por tanto, partimos de una población con soluciones aleatorias



# Fase inicial de recolección de recursos

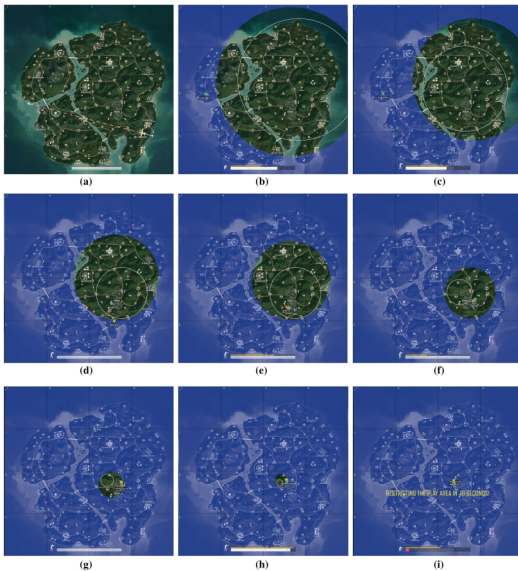
- Los jugadores aparecen en el mapa sin recursos
- En esta fase, buscan recursos por el mapa para ser más competitivos
- Esto se refleja en las búsquedas locales iniciales



## Fase final en la que el círculo se cierra

- A partir de cierto momento se cierra el mapa, muriendo los jugadores que queden fuera de la nueva zona
- Representamos esto haciendo un cierre no en el espacio, sino en el rango de *fitness* de los jugadores





# Batallas entre dos jugadores

- Los jugadores cuando se cruzan pelean entre sí
- Cuando dos jugadores están cerca (en el sentido de la distancia Manhattan) pelean entre sí
- El mejor equipado (mejor *fitness*) tiene más probabilidades de morir



# Jugadores que reviven aleatoriamente

- Cuando los jugadores mueren, tienen una pequeña probabilidad de resucitar <sup>1</sup>
- Cuando resucitan, pierden todos los recursos almacenados y aparecen en una posición aleatoria
- Tienen un periodo de gracia en el que tienen que recolectar recursos para ser competitivos y entrar dentro del círculo

Esto se ve reflejado de la forma:

- Cuando un jugador revive, se le asigna una posición (solución) aleatoria
- Tras revivir, tiene una búsqueda local para que recolecte recursos y entre al círculo si puede en su tiempo de gracia

---

<sup>1</sup>Esto no es común en los videojuegos de este género. Nos tomamos esta licencia para no perder variedad en la población, manteniendo una narrativa coherente ▶



# Decisiones basadas en la ingeniería

Introducimos algunas variaciones basadas en decisiones de ingeniería

- Uso de la distancia Manhattan en vez de la distancia euclídea, pues es más rápida de computar
- Mecánica del cierre del círculo basada en el fitness de los jugadores en vez del restringir zonas del espacio de búsqueda
- Tras hacer *parameter tuning*, la probabilidad de revivir no es baja, si no del 50%