

Del Píxel al Píxel

David Hernández Calvo
Escuela de Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
dahercal29@estudiantec.cr

Kevin Quesada Jiménez
Escuela de Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
kevin.qj@estudiantec.cr

Joshua Jiménez Leitón
Escuela de Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
jjimenez@estudiantec.cr

Resumen—En este artículo se presenta una solución que hace uso de un algoritmo genético para generar imitaciones de imágenes, dándoles una representación artística basada en técnicas reales de pintura y arte. A través de la experimentación, evaluamos la efectividad del algoritmo, para la imitación de imágenes artísticas. A partir de los experimentos realizados, los resultados demuestran la capacidad de la solución planteada para reproducir imágenes basadas en una imagen objetivo.

Index Terms—IA, Algoritmos Genéticos, Generación de Imágenes, Cubismo, Poligonal.

I. INTRODUCCIÓN

El presente artículo gira en torno a la implementación de un algoritmo que tiene como objetivo la generación de imágenes, que intentan aproximarse a una imagen objetivo y a su vez representarla en un estilo artístico, dicho estilo artístico en el que se basa el algoritmo será detallado posteriormente.

El diseño del algoritmo genético es fundamental para el funcionamiento de la solución. Por lo que en este artículo, se presentarán las decisiones clave de diseño, incluyendo los operadores genéticos utilizados y la generación de individuos. Nuestro trabajo también incluye una experimentación, diseñada para evaluar la efectividad del algoritmo. Presentaremos el diseño de los experimentos y el análisis de los resultados obtenidos. Los indicadores a tomar en cuenta serán: la calidad de la imitación generada, el tiempo de ejecución, la evolución a través de las generaciones y el mejor valor de *fitness* obtenido.

II. TÉCNICA ARTÍSTICA UTILIZADA

En la selección de la técnica artística a imitar, se ha optado por el estilo poligonal y el cubismo. Estos estilos artísticos, tienen en común que se caracterizan por la representación de objetos y escenas a través de figuras geométricas y poligonales, es por esto que ofrecen una estética que puede replicarse de gran forma mediante el uso de estos algoritmos.

La generación de imágenes basadas en figuras poligonales se adapta fácilmente a la manipulación de imágenes haciendo uso de la librería de OpenCV [1] para Python, lo que facilita el manejo y evolución del algoritmo.

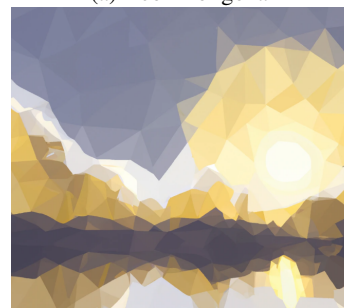
A continuación se explican y ejemplifican las técnicas seleccionadas.

A. Estilo Poligonal

La tendencia del diseño poligonal, incorpora elementos del cubismo al fusionar formas geométricas simples con



(a) Leon Poligonal



(b) Paisaje Poligonal

Fig. 1: Obras de Estilo Poligonal

superposiciones y efectos de transparencia. Lo que resulta particularmente notable en esta aproximación es su capacidad para generar una impresión de profundidad mediante la combinación de diversas formas y la creación de contrastes visuales [2].

Este estilo, con su énfasis en la representación a través de figuras geométricas simples, polígonos y contrastes, es ideal para ser manipulada y recreada por los procesos del algoritmo genético planteado.

En la figura 1 se muestran unos ejemplos de imágenes en estilo poligonal.

B. Cubismo

El movimiento artístico cubismo, tuvo su período de desarrollo desde 1907 hasta 1914 y sus principales creadores fueron Pablo Picasso y Georges Braque. Esta corriente se caracteriza por su representación de la realidad a través de elementos geométricos que resultan de un minucioso proceso



(a) Arlequin, Pablo Picasso (b) Autorretrato, Aristarj Lentúlov

Fig. 2: Obras de Estilo Cubismo

de análisis. Sin embargo, lo que distingue de manera más prominente al cubismo es su enfoque en la fragmentación de líneas y superficies, lo que da lugar a lo que se conoce como la perspectiva múltiple [3].

En la figura 2 se muestran unos ejemplos de imágenes en estilo de cubismo.

III. DISEÑO DEL ALGORITMO GENÉTICO

El diseño del algoritmo genético se compone de tres clases que desempeñan papeles importantes en el proceso de generación de las imágenes. En primer lugar, la clase **Individuo** se encarga de almacenar los datos importantes de cada espécimen generado, incluyendo su valor de *fitness* y su imagen correspondiente. Un aspecto fundamental en esta clase es el método de evaluación del *fitness*, que emplea la métrica *CIE 1976*. La elección de esta métrica se justifica por su capacidad para medir de manera efectiva las diferencias de color y percepción del ojo humano en imágenes, por lo que permite realizar la comparación entre las imágenes generadas y la imagen objetivo. Para llevar a cabo estos cálculos, se utiliza la biblioteca 'colour' de Python [4].

La clase **Genético** es el núcleo del algoritmo, pues aquí se realiza el procesamiento y la manipulación de las imágenes. Esta clase se encarga de llevar a cabo las etapas críticas del algoritmo genético, incluyendo la selección de individuos, la aplicación del cruce y la mutación, así como la ejecución de procesos para la generación de nuevos individuos.

Por último, la clase **Interfaz** proporciona una interfaz de usuario que permite elegir el valor de los parámetros clave del algoritmo, iniciar la ejecución y visualizar los resultados. Además, esta clase presenta los valores obtenidos en cada generación en forma gráfica, facilitando la comprensión y el análisis de la evolución de la imagen generada.

El algoritmo fue elaborado con Python usando las siguientes librerías:

- OpenCV (Generación y manipulación de imágenes)
- Colour (Uso de Métrica CIE 1976)
- Numpy (Cálculos y manipulación de vectores)

- Matplotlib (Generación de gráficos)
- Pillow (Manipulación de imágenes dentro de la interfaz)
- Imageio (Generación de GIF)
- Tkinter (Interfaz)

A. Hiperparámetros

Los hiperparámetros que recibe el algoritmo son los siguientes:

- Imagen objetivo
- Cantidad de individuos por generación
- Cantidad de generaciones que se crearan
- Cantidad de individuos que se toman para el método de selección
- Probabilidad de que ocurra una mutación
- Cada cuantas generaciones se mostrará el progreso en la animación final

B. Generación de Individuos

El método de generación de individuos se implementa en la clase **Individuo**. En este proceso, se crea una imagen que tiene las mismas dimensiones que la imagen objetivo, y se llena con un color sólido seleccionado al azar. Además, se generan entre 3 y 8 polígonos, cada uno de los cuales puede tener entre 3 y 6 vértices, y se asigna a cada polígono un color también aleatorio. El método se diseñó de esta forma para introducir una variabilidad significativa en cada individuo generado, garantizando que la generación no esté condicionada por la imagen objetivo desde el inicio.

C. Selección

El método de selección se encarga de determinar qué individuos actuarán como "padres" para la creación de nuevas generaciones mediante cruces o mutaciones. El método de selección implementado se lleva a cabo de la siguiente manera. Primero, se utiliza un hiperparámetro para especificar cuántos individuos serán seleccionados como posibles padres. Luego, se realiza una selección aleatoria entre los individuos de la población actual. A continuación, de entre los individuos seleccionados al azar, se elige el que posee el mayor valor de *fitness* como uno de los padres para el nuevo individuo. Este enfoque, conocido como selección por torneo, se justifica porque introduce un grado de aleatoriedad en la elección de los padres, lo que aumenta la diversidad y evita que la evolución se estanque en óptimos locales. Además al seleccionar al individuo con el mejor *fitness* de entre los candidatos, se busca que se transmitan características más deseables a las generaciones próximas.

D. Cruce

El método de cruce que implementamos se basa en la metodología desarrollada por Sebastian Charnot, tal como se describe en su repositorio [5]. Sin embargo, se adaptó este método a las bibliotecas y herramientas utilizadas, transitando de Pillow a OpenCV para realizar las manipulaciones de imágenes. La implementación del método de cruce consta de dos variantes, las cuales se eligen de acuerdo con ciertas

probabilidades calculadas en función del porcentaje restante del hiperparámetro de mutación:

- **Variante de Blend:** En esta variante, se realiza un proceso de fusión (*blend*) entre las imágenes que se cruzarán. Cada imagen contribuye al resultado final con un peso aleatorio. Esto significa que cada individuo padre influye de manera diferente en la composición del nuevo individuo, lo que introduce un poco más variabilidad en la descendencia.
- **Variante de Combinación Horizontal o Vertical:** La segunda variante del método de cruce consiste en combinar dos imágenes padres de una manera específica. Esto puede ser una combinación horizontal o vertical, y la elección entre ambas opciones se determina de forma aleatoria con una probabilidad del 50%. Además, el punto de división donde se realiza la combinación se elige al azar.

La probabilidad de que ocurra cada una de estas variantes de cruce se calcula en función del porcentaje restante del hiperparámetro de mutación. Esta estrategia nos pareció bastante adecuada para la solución. Pues al ofrecer dos variantes de cruce y seleccionar una de ellas de manera probabilística, se fomenta la diversidad y se evita que el proceso de cruces se vuelva predecible o repetitivo.

E. Mutación

El método de mutación en el algoritmo genético tiene una probabilidad de ocurrir establecida a través de un hiperparámetro. El proceso de mutación se realiza tomando la imagen del individuo padre seleccionado previamente y se modifica agregando entre 1 y 4 polígonos adicionales. Cada uno de estos polígonos puede tener entre 3 y 6 puntos, y se asigna un color aleatorio a cada uno de ellos. Esta estrategia de mutación busca introducir cambios y diversidad en el individuo mutado, pero al mismo tiempo busca mantener una estructura coherente con el estilo artístico general empleado.

F. Elitismo

El elitismo en la solución propuesta se implementa en los métodos de cruce. Su función es buscar que, incluso después de un cruce que no logre mejorar a los padres, se siga buscando la mejora en las generaciones futuras.

Inicialmente, se realiza el proceso de cruce entre los individuos seleccionados. Después de que se complete el cruce, se evalúa si el nuevo individuo generado es mejor que los padres en cuanto a *fitness*. Si los nuevos individuos no logran superar a los padres, se considera que el cruce no fue exitoso en términos de mejora, esto implica que se realiza una nueva selección de padres y se intenta nuevamente el proceso de cruce.

El general el uso del elitismo busca que no se pierda la mejora continua de la población en cada generación.

IV. EXPERIMENTACIÓN

Para la experimentación, se llevarán a cabo un total de seis pruebas, cada una con diferentes combinaciones de hiperparámetros del algoritmo genético. El objetivo es evaluar y

comparar el rendimiento y la eficacia del algoritmo en una variedad de configuraciones.

Para cada prueba, se registrará y analizará una serie de métricas significativas, incluyendo el resultado final de la imagen generada, el valor de *fitness* del mejor individuo encontrado, la representación gráfica de la evolución del promedio y el mejor *fitness* a lo largo de las generaciones, el tiempo de ejecución del algoritmo y una animación que recopilará todas las imágenes generadas durante el proceso evolutivo.

Estas pruebas nos permitirán comprender cómo las diferentes configuraciones de parámetros afectan el rendimiento del algoritmo y nos ayudarán a identificar las configuraciones más efectivas.

A. Diseño de Experimentos

En esta sección se describirán las pruebas diseñadas y la elección de hiperparámetros para cada tipo de prueba realizada. En total se realizaron 6 pruebas divididas en 3 tipos principales, esta división se realiza según las características de los hiperparámetros seleccionados.

1) *Pruebas 1 y 2 - Valores Equilibrados:* En las primeras dos pruebas, nuestro enfoque se centra en evaluar configuraciones de hiperparámetros equilibradas. Esto implica utilizar una cantidad considerable de individuos en la población y ejecutar un gran número de generaciones.

Además, mantenemos un porcentaje de mutación bajo, dado que la diversidad genética ya es significativa debido al tamaño de la población y la selección.

La combinaciones de parámetros son las siguientes:

- Cantidad de Individuos: 100 - 150
- Número de Generaciones: 10000 - 15000
- Cantidad de Seleccionados: 6 - 10
- Porcentaje de Mutación: 10 - 20

2) *Pruebas 3 y 4 - Baja Cantidad de Individuos, Altas Generaciones y Mutación:* Las pruebas 3 y 4 adoptan un enfoque diferente al reducir la cantidad de individuos en la población, pero manteniendo un número sustancial de generaciones. Esta configuración busca probar si, a pesar de contar con un número limitado de individuos en cada generación, el algoritmo puede lograr una aproximación de calidad con una mayor cantidad de generaciones. Para compensar la baja diversidad inicial, se utilizan porcentajes de mutación significativamente altos, con el fin de agregar variabilidad a la población. Además, se emplea una selección de individuos con valores bajos debido a la cantidad limitada de individuos.

La combinaciones de parámetros son las siguientes:

- Cantidad de Individuos: 25 -50
- Número de Generaciones: 15000 - 20000
- Cantidad de Seleccionados: 4 - 8
- Porcentaje de Mutación: 20 - 25

3) *Pruebas 5 y 6 - Gran Cantidad de Individuos, Pocas Generaciones y Baja Mutación:* En estas dos pruebas, se adopta un enfoque que usa una gran cantidad de individuos, pero limitando el número de generaciones. El objetivo es evaluar si la variabilidad de la población inicial, compuesta por



(a) La Noche Estrellada, Vincent van Gogh

Fig. 3: Obra utilizada como imagen objetivo de la experimentación

una cantidad significativa de individuos, puede conducir a un resultado aceptable en una cantidad reducida de generaciones. Dado que ya se cuenta con una diversidad considerable en la población inicial debido al gran número de individuos, se opta por mantener un porcentaje de mutación bajo, además, se utiliza una cantidad media de individuos seleccionados. La combinaciones de parámetros son las siguientes:

- Cantidad de Individuos: 200 - 250
- Número de Generaciones: 2500 - 5000
- Cantidad de Seleccionados: 6 - 10 Porcentaje de Mutación: 5 - 10

Finalmente es importante considerar que la imagen objetivo que se utilizará para estas pruebas, es una imagen a color y con dimensiones de 264 x 305 píxeles. La imagen es la obra conocida como *La Noche Estrellada* de Vincent van Gogh, presentada en la figura 3.

Hemos seleccionado esta obra ya que es lo suficientemente compleja para proponer un desafío al algoritmo y que los resultados sean significativos, dada la combinación de formas y colores presentes en la imagen.

B. Resultados de los Experimento

En esta sección se presentarán los resultados de cada prueba específica, a excepción del archivos GIF, pues estos estarán disponibles en el repositorio del proyecto que se adjunta en el anexo del presente documento.

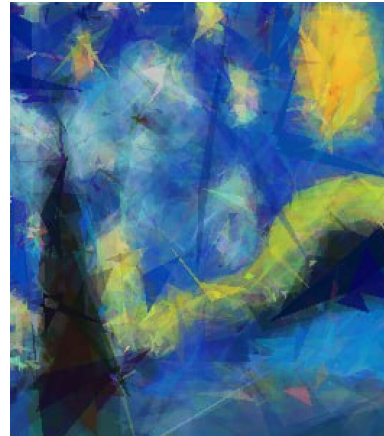
1) **Prueba 1:** Esta prueba pertenece al primer grupo con valores equilibrados.

Duración: 3 hrs y 13 mins

Parámetros de la prueba:

- Cantidad de Individuos: 100
- Número de Generaciones: 10000
- Cantidad de Seleccionados: 6
- Porcentaje de Mutación: 10
- Mostrar cada n Generaciones: 100

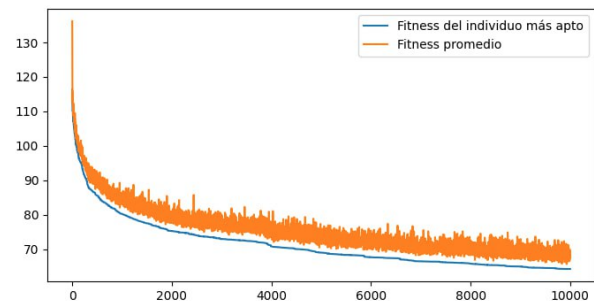
Resultado final:



El mejor fit de la generación 10000 tiene un *fitness* de: 64.2180596012

Gráfico de *fitness*:

En el gráfico se denota cómo va mejorando el *fitness* del mejor individuo por generación de manera acelerada en las primeras 2000 generaciones. Durante el resto de generaciones la mejora es poca de generación a generación pero vemos que el porcentaje de mutaciones no es suficiente como para desequilibrar para mejora o deterioro del mejor individuo. En cuanto al promedio, se ve como sí hay picos leves pero tiene a la baja, de la mano con el mejor individuo por generación.



2) **Prueba 2:** Esta prueba pertenece al primer grupo con valores equilibrados.

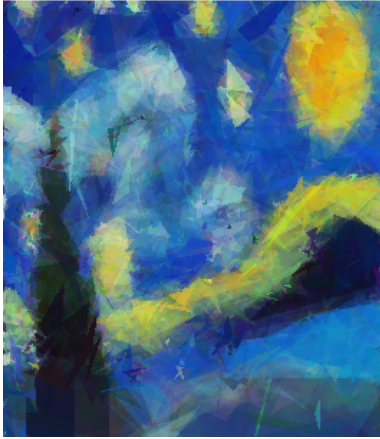
Duración: 12 hrs y 6 mins

Parámetros de la prueba:

- Cantidad de Individuos: 150
- Número de Generaciones: 15000
- Cantidad de Seleccionados: 10

- Porcentaje de Mutación: **20**
- Mostrar cada n Generaciones: **100**

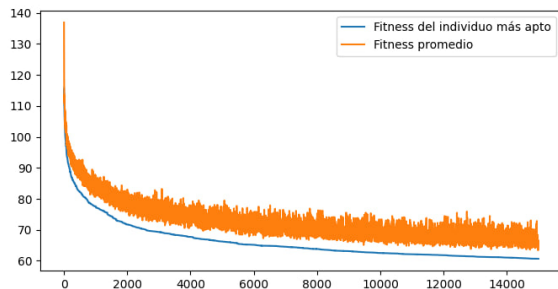
Resultado:



El mejor fit de la generacion 15000 tiene un *fitness* de:
60.6766447677

Gráfico de *fitness*:

A pesar de tener un 50% más de generaciones y de individuos, vemos que el comportamiento que tuvieron en el pasar de las generaciones, tomó bastante tiempo y, a pesar que sí se percibe una mejoría respecto a la prueba 1, esta no fue tan significativa.



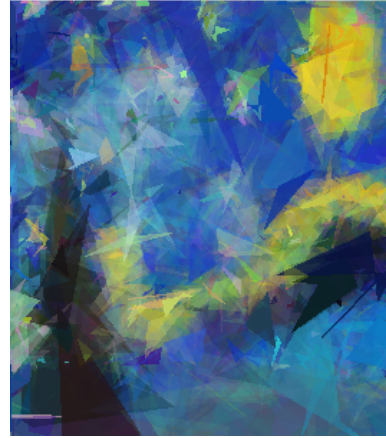
3) **Prueba 3:** Esta prueba pertenece al segundo grupo con baja cantidad de individuos, altas generaciones y mutación.

Duración: 3 hrs y 29 mins

Parámetros de la prueba:

- Cantidad de Individuos: **25**
- Número de Generaciones: **15000**
- Cantidad de Seleccionados: **4**
- Porcentaje de Mutación: **20**
- Mostrar cada n Generaciones: **100**

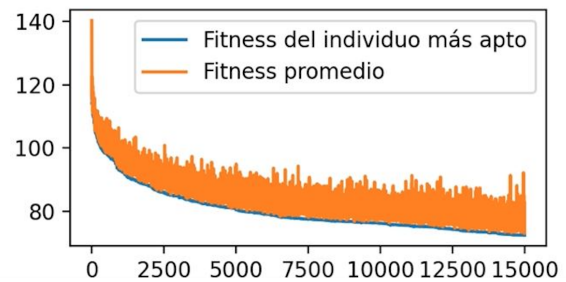
Resultado:



El mejor fit de la generacion 15000 tiene un *fitness* de:
72.4867271583

Gráfico de *fitness*:

Para esta prueba en particular, tenemos que la gran cantidad de generaciones no necesariamente significa una mejora en el resultado. De hecho, a partir de la generación 6000, el progreso de los resultados es mínima entre generación en generación es mínima.



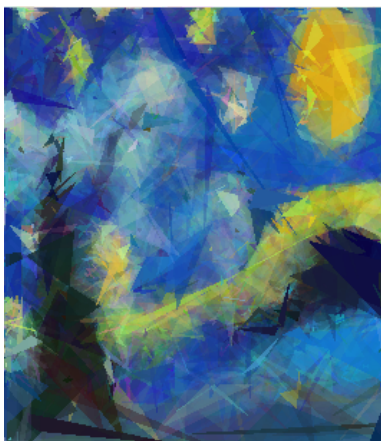
4) **Prueba 4:** Esta prueba pertenece al segundo grupo con baja cantidad de individuos, altas generaciones y mutación.

Duración: 5 hrs y 24 mins

Parámetros de la prueba:

- Cantidad de Individuos: **50**
- Número de Generaciones: **20000**
- Cantidad de Seleccionados: **8**
- Porcentaje de Mutación: **25**
- Mostrar cada n Generaciones: **100**

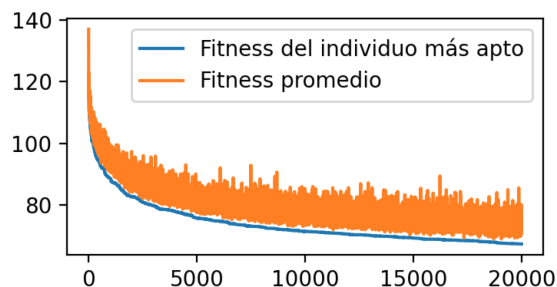
Resultado:



El mejor fit de la generacion 20000 tiene un *fitness* de:
67.2851596393

Gráfico de *fitness*:

Para la prueba 4, vemos que al tener menos individuos, le tomó una mayor cantidad de generaciones para poder mejorar. El progreso entre generación y generación es considerable y requirió de una vasta cantidad de generaciones para obtener un resultado moderado, pues otras pruebas consiguieron resultados con mayor precisión.



5) **Prueba 5:** Esta prueba pertenece al tercer grupo con gran cantidad de individuos, pocas generaciones y baja mutación.

Duración: 2 hrs y 40 mins

Parámetros de la prueba:

- Cantidad de Individuos: **200**
- Número de Generaciones: **2500**
- Cantidad de Seleccionados: **6**
- Porcentaje de Mutación: **5**
- Mostrar cada n Generaciones: **100**

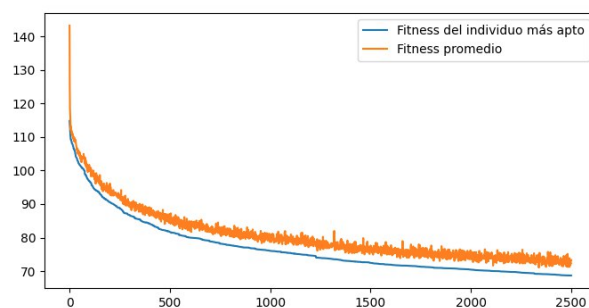
Resultado:



El mejor fit de la generacion 2500 tiene un *fitness* de:
68.6886820452

Gráfico de *fitness*:

A pesar de tener pocas generaciones, la duración es considerable y es perceptible que la mejora se incrementa de manera más acelerada en comparación con las pruebas con muchas generaciones y menos individuos.



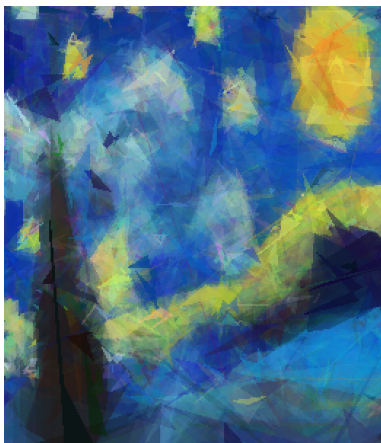
6) **Prueba 6:** Esta prueba pertenece al tercer grupo con gran cantidad de individuos, pocas generaciones y baja mutación.

Duración: 8 hrs y 6 mins

Parámetros de la prueba:

- Cantidad de Individuos: **250**
- Número de Generaciones: **5000**
- Cantidad de Seleccionados: **10**
- Porcentaje de Mutación: **10**
- Mostrar cada n Generaciones: **100**

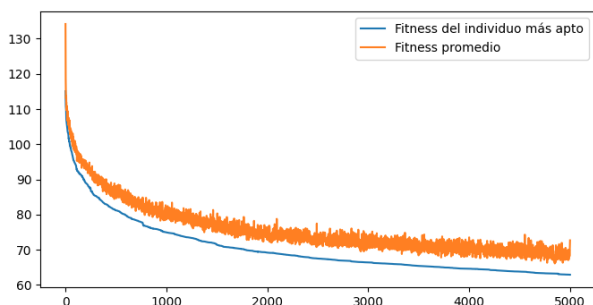
Resultado:



El mejor fit de la generacion 5000 tiene un *fitness* de:
62.9186477442

Gráfico de *fitness*:

La prueba 6 comparada a la prueba 5 inicia con parámetros un poco más flexibles, incluso incrementando la mutación y el doble de generaciones. Esto nos lleva a una prueba con resultados destacables por encima de los demás y permite ver que una buena variedad de características y de individuos fomenta un mejor resultado.



V. ANÁLISIS DE LOS RESULTADOS

El análisis de los resultados revela una serie de patrones y tendencias en el rendimiento del algoritmo genético con diferentes configuraciones:

1) *Virtudes con muchos individuos*: Las pruebas con muchos individuos y pocas generaciones demostraron que, si bien la creación de cada generación llevó más tiempo debido al alto número de individuos, estas configuraciones necesitaron menos generaciones en total para alcanzar resultados óptimos.

2) *Dificultad con pocos individuos*: En contraste, las pruebas con un número limitado de individuos inicialmente crearon generaciones más rápidamente, pero con el tiempo se volvieron menos efectivas. El efecto del elitismo, debido a que hay pocas opciones de cruce, ocasiona que se requieran múltiples intentos.

3) *Balance óptimo*: Las pruebas que utilizaron una cantidad equilibrada de individuos y generaciones lograron resultados sobresalientes, manteniendo un equilibrio entre el tiempo de ejecución y el *fitness* final. Esto sugiere que una combinación adecuada de individuos y generaciones permite encontrar mejores resultados.

4) *Impacto de la cantidad de individuos*: Se observó que un mayor número de individuos por generación condujo a un mejor *fitness* promedio y una menor variación entre generaciones, como se reflejó en los gráficos de evolución. Esto indica que una mayor diversidad inicial y una población más grande pueden ser beneficiosas.

5) *Mejores configuraciones*: Las pruebas 2 y 6 demostraron ser las más efectivas en términos de *fitness*, aunque requirieron más tiempo de ejecución. Las pruebas 5 y 2 también generaron resultados aceptables en tiempos más reducidos, lo que sugiere que podrían ser las configuraciones óptimas para encontrar un equilibrio entre calidad y eficiencia.

6) *Impacto del hiperparámetro*: Se concluye que el hiperparámetro que más influyó en el rendimiento fue la cantidad de individuos por generación, ya que esto afectó significativamente la diversidad, la velocidad y la calidad de las soluciones finales.

VI. REFERENCIAS BIBLIOGRÁFICAS

[1] OpenCV. “OpenCV: OpenCV-Python Tutorials”. OpenCV documentation index. Accedido el 25 de septiembre de 2023. [En línea]. Disponible: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

[2] A. Acebal. “La tendencia del diseño gráfico poligonal y el uso de figuras geométricas”. Graficatessen – Blog de diseño gráfico, ilustración y creación visual. Accedido el 25 de septiembre de 2023. [En línea]. Disponible: <https://graficatessen.es/la-tendencia-del-diseno-grafico-poligonal-y-el-uso-de-figuras-geometricas/>

[3] IFEMA MADRID. “¿Qué es el Cubismo? Artistas y cuadros famosos”. IFEMA MADRID: Eventos, Ferias y Congresos. Accedido el 25 de septiembre de 2023. [En línea]. Disponible: <https://www.ifema.es/noticias/arte/que-es-el-cubismo>

[4] Colour. “Colour Difference — Colour 0.4.3 documentation”. 1 Draft Release Notes — Colour 0.4.3 documentation. Accedido el 25 de septiembre de 2023. [En línea]. Disponible: <https://colour.readthedocs.io/en/develop/colour.difference.html>

[5] S. Charnot. “Genetic-Algorithm-Image-Recreation”. GitHub. Accedido el 25 de septiembre de 2023. [En línea]. Disponible: <https://github.com/SebastianCharnot/Genetic-Algorithm-Image-Recreation>

A. Anexo

Link del repositorio de git:

<https://github.com/Kevin-197/Proyecto1-IA>