

ALGORITMOS GENÉTICOS APLICADOS A LA RECONSTRUCCIÓN DE IMÁGENES

Juan Pablo Luna Felipez, Ph.D.
jplunaf@gmail.com
Ingeniería Informática
Universidad Nacional “Siglo XX”
Llallagua – Bolivia

Resumen

Con los avances en tecnología, sobre todo en los sistemas de cómputo y debido a los nuevos problemas que surgen a medida que evolucionan las necesidades y avanza la ciencia y la tecnología, han surgidos nuevos métodos orientados a la inteligencia artificial los cuales buscan imitar la inteligencia humana, uno de estos son los algoritmos genéticos que se enmarcan en el campo de la computación evolutiva.

La computación evolutiva es una rama de la inteligencia artificial inspirada en los mecanismos de evolución biológica de la teoría darwiniana, en el que uno de sus paradigmas son los algoritmos genéticos.

Los Algoritmos Genéticos son un tipo de algoritmo de búsqueda heurística inspirado en la genética y la selección natural donde los individuos más aptos de una población son los que sobreviven al adaptarse más fácilmente a los cambios que se producen en su entorno.

Dentro de este campo existen diversos estudios y aplicaciones que buscan resolver problemas de optimización, uno de esos campos de aplicaciones es en el tratamiento de imágenes.

El presente artículo aborda justamente la reconstrucción del escudo de la carrera desde una imagen bidimensional empleando algoritmos genéticos, como un proceso de acercamiento a los algoritmos genéticos aplicados al tratamiento de imágenes.

Palabras Clave: Algoritmos Genéticos, Computación Evolutiva, Inteligencia Artificial, Mutación, Reconstrucción de Imágenes.

Abstract

With advances in technology, especially in computing systems, and due to new problems that arise as needs evolve and science and technology advance, new methods oriented toward artificial intelligence have emerged, which seek to imitate intelligence. . human, one of these are genetic algorithms that are part of the field of evolutionary computing. Evolutionary computing is a branch of artificial intelligence inspired by the mechanisms of biological evolution of Darwinian theory, in which one of its paradigms is genetic algorithms.

Genetic Algorithms are a type of heuristic search algorithm inspired by genetics and natural selection where the fittest individuals in a population are those that survive by adapting more easily to the changes that occur in their environment.

Within this field there are various studies and applications that seek to solve optimization problems, one of these fields of applications is in image processing.

This article addresses precisely the reconstruction of the race shield from a two-dimensional image using genetic algorithms, as a process of approaching genetic algorithms applied to image processing.

Keywords: Artificial Intelligence, Genetic Algorithms, Evolutionary Computation, Image Reconstruction, Mutation

1. INTRODUCCIÓN

La computación evolutiva es una rama de la inteligencia artificial inspirada en los mecanismos de evolución biológica de la teoría darwiniana. Dentro de este campo, uno de sus paradigmas más prominentes son los algoritmos genéticos.

Estos algoritmos representan un tipo de búsqueda heurística inspirada en la genética y la selección natural, donde los individuos más aptos de una población son aquellos que sobreviven al adaptarse más fácilmente a los cambios en su entorno.

Este artículo explora el uso de algoritmos genéticos para la reconstrucción de imágenes, específicamente aplicados al escudo de una carrera universitaria.

Computación Evolutiva

De acuerdo con (Sanchez, 2012) la Computación evolutiva “es una rama de la inteligencia artificial, inspirada en los mecanismos de evolución biológica, que involucra problemas de optimización combinatoria. Está considerada como una técnica metaheurística y por lo tanto es adecuada para la resolución de problemas con espacios de búsqueda extensos, dispersos, no lineales y no convexos, en donde otros métodos no son capaces de encontrar soluciones en un tiempo razonable”

Paradigmas de Computación Evolutiva

La Computación evolutiva tiene cuatro paradigmas fundamentales que de acuerdo con (Sanchez, 2012) son:

Los Algoritmos Genéticos: Fueron desarrollados por John H. Holland y sus colaboradores. Utilizaron inicialmente la codificación binaria, aunque en la actualidad también se ha extendido a la codificación con números reales. Emplea operadores genéticos de selección, recombinación y mutación teniendo mayor relevancia los operadores de selección y recombinación.

La Programación Evolutiva: Fue desarrollada por Lawrence J. Fogel con la idea de simular la evolución como medio de aprendizaje con el fin de obtener una inteligencia artificial. Esta trabaja directamente con las variables de diseño y los nuevos individuos son generados a partir de un solo padre mediante el uso exclusivo de la mutación. En cierto modo es una variación de los Algoritmos Genéticos con reproducción asexual.

Las Estrategias Evolutivas: Fueron desarrolladas por Ingo Rechenberg, Hans-Paul Schwefel y sus colaboradores. Utilizan generalmente la codificación con números reales.

La función de codificación transcribe dos tipos de variables: las variables objeto y las estratégicas. Las variables objeto se corresponden con las del problema que se desea resolver mientras que las estratégicas son los parámetros mediante los cuales se gobierna el proceso evolutivo. Emplea operadores genéticos de selección y mutación.

La Programación Genética: Fue desarrollada por Michael Cramer y John Koza con la idea de generar automáticamente programas de computador en diferentes lenguajes y formas como los árboles de decisión y grafos empleando operadores genéticos para tal fin.

Algoritmos Genéticos

Antecedentes

Los Algoritmos Genéticos fueron desarrollados por John Henry Holland y sus colaboradores entre los que destaca De Jong de la Universidad de Michigan a finales de los sesenta. Después de estudiar un libro escrito por Fisher titulado “La teoría genética de la selección natural”, Holland aprendió que la evolución es una forma de adaptación más potente que el simple aprendizaje y tomó la decisión de aplicar estas ideas para desarrollar un algoritmo que permitía a los ordenadores imitar el proceso de la evolución.

Posteriormente, en 1989, Goldberg que también fue discípulo de Holland publicó un libro que le dio una fuerte base científica, citando no menos de 73 aplicaciones exitosas de los algoritmos genéticos (Sanchez, 2012).

Definición

De acuerdo con (Sanchez, 2012) los Algoritmos Genéticos son un tipo de algoritmo de búsqueda heurística inspirados en la genética y la selección natural enunciada por el naturalista inglés Charles Darwin en el libro “The Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life” (El Origen de las Especies). Según esta teoría los individuos más aptos de una población son los que sobreviven al adaptarse más fácilmente a los cambios que se producen en su entorno.

(Santos & Tundo, 2004) define a los algoritmos genéti-

cos como “métodos sistemáticos para la resolución de problemas de búsqueda y optimización que aplican a estos los mismos métodos de la evolución biológica: selección basada en la población, reproducción y mutación”.

Elementos de los algoritmos genéticos

Población

La población es un conjunto de individuos que representan el conjunto de soluciones evaluadas durante una generación (iteración). Idealmente la primera población debería estar formada por individuos contenidos en todo el espacio de búsqueda.

Los individuos representan una de las soluciones del problema planteado. Dada la dualidad existente entre el genotipo y el fenotipo de los individuos estas soluciones también lo son. La solución genotípica viene representada por los cromosomas, que es con lo que trabajan los Algoritmos Genéticos.

Por otro lado, la solución fenotípica representa el modelo o la estructura en que la solución genotípica se decodifica.

Genes

Es una porción del cromosoma que generalmente codifica el valor de un solo parámetro acotado dentro de un cierto rango o dominio. La cantidad de valores que puede tomar un parámetro suele ser discreta y debe tenerse en cuenta que a mayor número más complejo tiende a ser el espacio de búsqueda.

La estructura del gen depende de la función de codificación o mapeo empleada.

Función de aptitud

De acuerdo con (Sanchez, 2012) la adaptación de cada individuo de una población al medio se realiza mediante la función de aptitud o fitness. Para cuantificar esta adaptación o fitness primero debe decodificarse el cromosoma, ya que esta función está definida en el espacio fenotípico.

El grado de adaptación de un individuo será el que determine su probabilidad de reproducción o incluso su eliminación de la población.

De acuerdo con (Llorente, 2012) la función que calcula de cuánta calidad es un individuo es fundamental para el correcto funcionamiento del algoritmo, ya que guía la evolución de la población

Operadores de Selección

De acuerdo con (Dominioni & Musso, 2003) los operadores de selección más importantes son:

Selección Proporcional (Roulette Wheel): Tipo más común de selección. Consiste en seleccionar individuos de manera proporcional a su fitness

Cada individuo tendrá una probabilidad p de ser seleccionado que será calculada como el fitness de ese individuo dividido la suma de los fitness de todos los individuos de la población.

De esta forma los individuos con mejor fitness tendrán mayor probabilidad de ser seleccionados.

Selección Uniforme: En este tipo de selección todos los individuos tienen la misma probabilidad de ser seleccionados. Se eligen aleatoriamente los individuos.

Torneo: Se seleccionan por algún otro método de selección N individuos. De esos N individuos sobreviven los K mejores (los K que posean mejor fitness).

Rango: Se ordenan los individuos según su fitness de menor a mayor. Se le asigna a cada individuo un rango que depende de su posición en la lista ordenada (peor individuo rango menor). La probabilidad de que un individuo sea seleccionado va a ser proporcional a su rango o sea la probabilidad de un individuo y de ser seleccionado es el rango de ese individuo dividido la suma de rangos del resto de los individuos de esa población.

Cruzamiento:

De acuerdo (Dominioni & Musso, 2003) los operadores de cruzamiento son operadores de cardinalidad dos es decir que se necesitan dos individuos para que pueda ser aplicado. Además estos operadores retornan dos individuos. A continuación distinguiremos algunos tipos de cruzamiento.

Cruzamiento de un punto

El cruzamiento de un punto es el tipo más común de cruzamiento. En él dado una codificación de individuos establecida por un cromosoma de largo L se selecciona un número aleatorio N entre 1 y $L-1$. El primer nuevo individuo estará dado por las primeras N posiciones del cromoso-

ma 1 y las L-N últimas posiciones del cromosoma 2.

El segundo nuevo individuo estará dado por las primeras N posiciones del cromosoma 2 y las L-N últimas posiciones del cromosoma 1.

A partir de la combinación de estos cromosomas salen los dos nuevos individuos.

Aquí se presenta un ejemplo para clarificar la idea:

Sean los Strings:

```
100000000011
001111011011
```

El largo de cada String es 12. Eligiendo aleatoriamente un número entre 1 y 11.

Se obtiene el 3.

```
100|000000011
001|111011011
```

Los Strings finales serán los siguientes:

```
100111011011
001000000011
```

Cruzamiento de dos puntos:

En este tipo de cruzamiento se escogen dos números al azar entre 1 y L-1 siendo L el largo de los individuos. Estos intercambian el segmento que cae entre esos dos puntos.

Ejemplo:

Supongamos que se tiene los números aleatorios 4 y 8 en un String de largo 12.

```
1001|1101|1011
0010|0000|0011
```

Intercambio los segmentos se tiene:

```
100100001011
001011010011
```

Cruzamiento de n puntos:

Es una generalización del tipo de cruzamiento. Se escogen puntos al azar y se intercambian

segmentos provenientes del punto de corte impar al punto de corte par próximo. Se puede ver el String como un anillo de manera que al llegar al último valor se comienza de nuevo desde la izquierda.

Por ejemplo cruzamiento de 5 puntos:

```
1|00|111|0|1101|1
0|01|000|0|0001|1
0|01|111|0|1101|1
1|00|000|0|0001|1
```

Cruzamiento uniforme:

El cruzamiento uniforme funciona de la siguiente manera. Tenemos dos Strings a los cuales les voy a aplicar este cruzamiento. L es el largo de cada String. Por cada posición entre 1 y L escoger aleatoriamente de cuál de los dos padres voy a escoger su valor genético en esa posición.

Ejemplo:

```
Pa:100101
Pb:001101
```

Eligiendo aleatoriamente el valor de que individuo va a tener el String en cada posición se obtiene:

```
PbPbPaPaPbPa:
000101.
```

Mutación

De acuerdo (Dominioni & Musso, 2003) este es un operador que se aplica sobre un solo individuo y te retorna un solo individuo.

El tipo más común de mutación es aquel en el que para cada individuo se recorren todos sus valores genéticos para cada posición y se tira una moneda con probabilidad pm para ver si el valor en esa posición cambia o no.

De cambiar el valor cambia por su complementario. Por ejemplo en el caso de una codificación de bits cambia un 1 por un 0 y un 0 por un 1.

Operadores de ordenamiento:

De acuerdo (Dominioni & Musso, 2003) los operadores de ordenamiento son utilizados en lugar de cruzamiento para problemas donde se utiliza inversión y la

codificación de cada individuo establece una dependencia entre la posición dentro del String y su valor. O sea, cada componente dentro del String representa un tipo específico de información y no queremos que al combinar dos Strings desaparezca un tipo de información.

Al aplicar inversión se desordenan las posiciones del String y se puede perder información al aplicar el cruzamiento. Veamos el siguiente ejemplo:

```
A= 1 2 3 4 | 5 6 7 8
    1 0 1 1 | 1 0 1 1
B= 1 2 6 5 | 4 3 7 8
    1 0 0 1 | 1 1 1 1
```

Al realizar el cruzamiento se obtiene:

```
A'= 1 2 3 4 | 4 3 6 8
     1 0 1 1 | 1 1 1 1
B'= 1 2 6 5 | 5 6 7 8
     1 0 0 1 | 1 0 1 1
```

De esta forma nos damos cuenta que el String A no posee información sobre la posición 5 y posee dos veces información sobre la posición 4. Esto hace que el operador de cruzamiento no sea un operador adecuado para estas circunstancias.

Por lo tanto se han utilizado otros tipos de combinaciones entre strings que se mencionan a continuación:

a) PMX:

Bajo PMX dos strings son alineados, y dos lugares de cruzamiento son escogidos uniformemente a través de los Strings.

Estos dos puntos de cruzamiento definen la región de mapeo, que es utilizada para efectuar un cruzamiento posición por posición a través de los operadores de intercambio.

Ejemplo:

```
A= 9 8 4 | 5 6 7 | 1 3 2 10
B= 8 10 4 | 2 3 10 | 9 5 4 6
```

Lo que hago es mapear el String B con el A, o sea el 5 con el 2 el 6 con el 3 y el 7 con el 10 cambiando sus lugares.

```
A'= 9 8 4 | 2 3 10 | 1 6 5 7
B'= 8 10 1 | 5 6 7 | 9 2 4 3
```

b) OX (Order CrossOver):

Este tipo de operador trabaja similar al PMX. Primero selecciono las regiones de mapeo.

```
A = 9 8 4 | 5 6 7 | 1 3 2 10
B = 8 10 4 | 2 3 10 | 9 5 4 6
```

En lugar de utilizar intercambios punto a punto como hace PMX lo que hacemos es utilizar un deslizamiento para llenar los huecos dejados por transferir las posiciones mapeadas.

Por ejemplo cuando el String B mapea con el String A los puntos 5 6 y 7 dejarán huecos (marcados con H)

```
B = 8 H 1 | 2 3 10 | 9 H 4 H
```

Luego los huecos son desplazados.

```
B = 2 3 10 | H H H | 9 4 8 1
```

Y son llenados obteniendo los siguientes resultados.

```
A' = 5 6 7 | 2 3 10 | 1 9 8 4
B' = 2 3 10 | 5 6 7 | 9 4 8 1
```

c) CX (Cycle CrossOver):

Este tipo de cruzamiento es diferente.

En lugar de escoger los lugares de cruzamiento, comenzamos en la parte izquierda del primer padre.

Suponemos que tenemos

```
C = 9 8 2 1 7 4 5 10 6 3
D = 1 2 3 4 5 6 7 8 9 10
C' = 9 _ _ _ _ _
```

Nos fijamos en el padre D, vemos que al nueve le corresponde el 1 entonces colocamos en C' el 1.

```
C' = 9 _ _ 1 _ _ _
```

Así sucesivamente hasta que se termina el ciclo.

```
C' = 9 _ _ 1 _ 4 _ 6 _
```

Luego de la finalización del primer ciclo los restantes lugares son llenados a través del otro String. Completando el ejemplo obtengo:

```
C' = 9 2 3 1 5 4 7 8 6 10
D' = 1 8 2 4 7 6 5 10 9 3
```

Componentes de un algoritmo genético

De acuerdo con (Sanchez, 2012) cualquier Algoritmo Genético, independientemente del problema planteado, está compuesto por los siguientes componentes:

Una representación de las potenciales soluciones del problema.

Un método para crear la población inicial de posibles soluciones.

Una función de evaluación que juega el papel de medio calificando las soluciones en base a su fitness.

Operadores genéticos que alteran la composición de los hijos.

Algoritmo Genético Simple

De acuerdo (Dominioni & Musso, 2003) fue propuesto por Golberg en 1987 y a partir del modelo original de Holland, en donde la idea del mismo es justamente a partir de una población inicial (generada aleatoriamente) aplicar los operadores de selección y recombinación para generar nuevos individuos pertenecientes a una nueva población y realizar esto en forma iterativa.

El esquema general del mismo es el siguiente:
P,MP: Población y: generación $t=0$

Inicializar (P(t));

Mientras No (Condición de Finalización(P(t)))

{

Evaluar(P(t));

MP(t)=Selección(P(t)); MP(t)=Cruzamientos(MP(t));
P(t+1)=Mutaciones(MP(t)); $t=t+1$

}

Donde se destacan las tres etapas principales:

Selección: Luego de evaluar a los individuos de la población actual, se realiza un mecanismo de selección, en donde los individuos elegidos formarán parte de la población temporal MP(t)

Recombinación: En los algoritmos genéticos los operadores de recombinación están integrados por el operador de Cruzamiento y el opera-

dor de Mutación.

Reemplazo: En el AGS el reemplazo es generacional, por lo tanto una nueva población reemplaza completamente a la población anterior (en otros modelos de AG se establecen determinadas políticas de reemplazo)

Algoritmo de reconstrucción de imágenes empleando algoritmos genéticos.

2. MATERIALES Y MÉTODOS

El estudio emplea un enfoque cuantitativo y experimental para evaluar la efectividad de los algoritmos genéticos en la reconstrucción de imágenes bidimensionales. La investigación se centra en la reconstrucción del escudo de una carrera universitaria, utilizando algoritmos genéticos como herramienta de optimización. Para este estudio, se implementó el algoritmo propuesto por Roger Johansson Alsing, que de acuerdo con (Lopez, 2015) construyó un algoritmo que permite empleando algoritmos genéticos reconstruir una imagen empleando polígonos semitransparentes, cuyos pasos son:

Defínase una cadena de ADN al azar (inicio de la aplicación)

Copíese la secuencia actual del ADN y mútese ligeramente

Use la nueva cadena de ADN para renderizar los polígonos

Compárese la imagen contra la imagen fuente.

Si la nueva imagen se ve más parecida a la imagen fuente que la anterior, entonces sobre escribese el ADN actual por el nuevo ADN

Repita desde el primer paso.

Para este propósito se construyó un prototipo que se codifican empleando el lenguaje C# y el entorno de desarrollo integrado Visual Studio.

3. RESULTADOS

Los resultados obtenidos fueron los siguientes:

Inicialmente se probó a construir un rombo de fondo Blanco y el objeto en color naranja los resultados de reconstrucción a partir de 40 polígonos fueron altamente satisfactorios como se puede apreciar en la figura:

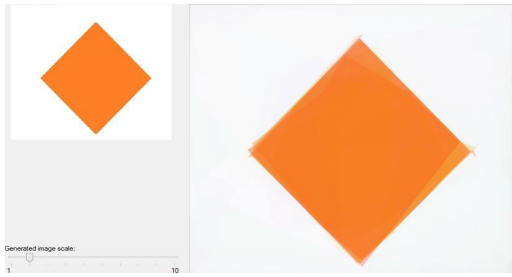


Figura 1: Reconstrucción de un Rombo
Fuente: Elaboración Propia

Seguidamente se reconstruyo un pentágono de fondo Blanco, el contorno del objeto en color negro y el fondo del objeto en color naranja los resultados de reconstrucción a partir de 40 polígonos fueron satisfactorios con la excepción de que el algoritmo apenas realiza una pequeña aproximación a la reconstrucción del contorno del objeto como se puede apreciar en la figura:

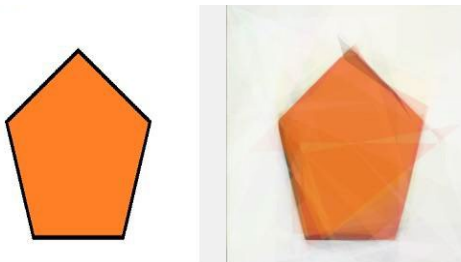


Figura 2: Reconstrucción de un Pentágono
Fuente: Elaboración Propia

Para terminar se construyó el escudo de la carrera Ingeniería Informática que es un objeto compuesto por varias figuras geométricas y que no tenía colores en un solo tono, el cual además tenía un fondo blanco, donde por la los resultados de reconstrucción a partir de 60 polígonos fueron medianamente satisfactorios como se puede apreciar en la figura

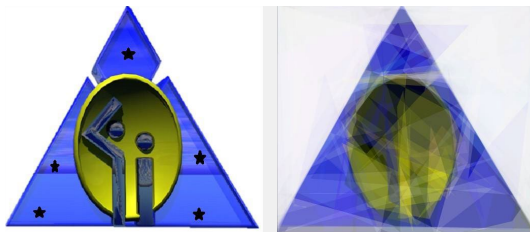


Figura 6: Reconstrucción del escudo de la carrera Ingeniería Informática de la Universidad Nacional "Siglo XX"
Fuente: Elaboración Propia

Es necesario mencionar que si bien las pruebas fueron satisfactorias, sin embargo el tiempo de reconstrucción de cada una es lento, tomando varios minutos por imagen, lo que implica que el algoritmo debe ser mejorado.

CONCLUSIONES

Al culminar el presente estudio, principalmente se puede concluir:

El algoritmo de Alsing permite reconstruir figuras geométricas a partir de una imagen empleando polígonos semitransparentes y reconstruye figuras geométricas con resultados medianamente satisfactorios.

El algoritmo tiene un tiempo de reconstrucción muy elevado, haciendo el proceso lento y tomando varios minutos por imagen, lo que implica que el algoritmo debe ser mejorado.

Se logró reconstruir el escudo de la carrera pero con resultados medianamente satisfactorios.

REFERENCIAS

Dominioni, F., & Musso, P. (2003). Algoritmos Genéticos Incrementales.

Llorente, M. (2012). Programación Genética en Mercados Financieros. Barcelona: Universidad Politécnica de Cataluña.

Lopez, M. (2015). Un algoritmo genético que pinta a la Mona Lisa. Recuperado el 28 de Octubre de 2017, de unocero: <https://www.unocero.com/noticias/ciencia/un-algoritmo-genetico-que-pinta-a-la-mona-lisa/>

Sanchez, S. (2012). Optimización Estructural y Topológica de Estructuras Morfológicamente no definidas mediante Algoritmos Genéticos. Valencia: Universidad Politécnica de Valencia.

Santos, & Tudo. (2004). Un Algoritmo Genético para la Restauración de Imágenes Digitales