Reconstrucción de Imágenes con Algoritmos Evolutivos

Andrés Matesanz

Universidad de Málaga,
Bulevar Louis Pasteur, 35, 29071 Málaga, Spain

Resumen Los algoritmos evolutivos tratan de replicar la evolución natural con el objetivo de encontrar soluciones óptimas a problemas con un espacio de búsqueda demasiado amplio como para aplicar técnicas de fuerza bruta. En este informe se presenta un análisis sobre la aplicación de algoritmos evolutivos a la reconstrucción de imágenes.

1. Introducción

En matemáticas y ciencias de la computación, un problema de optimización es aquél en el que se trata de encontrar la mejor solución dentro de todas las soluciones factibles a dicho problema. Los problemas de optimización con múltiples variables discretas (o continuas si se discretiza su espacio) pertenecen a los conocidos como problemas de optimización combinatoria. La solución a un problema de optimización es tal que ninguna otra solución dentro del espacio de búsqueda minimiza más la función objetivo. Explorar todo el espacio de búsqueda, en general, es altamente costoso en cuanto a recursos de tiempo y computación, por ello se emplean algoritmos de aproximación conocidos como metaheurísticas.

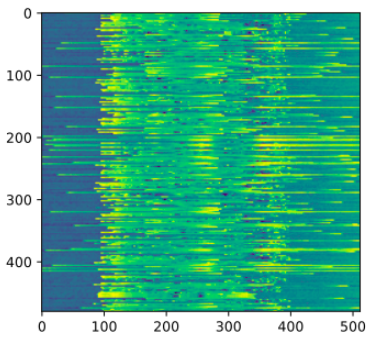


Figura 1: Imagen desordenada a lo largo del eje y.

Los algoritmos evolutivos son un subconjunto de metaheurísticas, dentro de los problemas de optimización, inspirados en la evolución natural de las especies. Se genera un número determinado de individuos, llamado población, con configuraciones distintas entre sí, cuyo objetivo común es minimizar la función objetivo dentro del espacio de búsqueda. Cada población nace de su predecesora, aplicando una serie de técnicas de cruce y mutación a aquellos individuos que mejores resultados hayan demostrado en cada iteración.

En este caso concreto se va a tratar de aplicar un algoritmo evolutivo a la reconstrucción de una imagen a lo largo de su eje y. Para ello se presentan una serie de imágenes desordenadas cuyas columnas de píxeles se encuentran en la posición correcta. pero cuyas filas no, podemos ver un ejemplo en la imagen 1.

2. Ejecución

La aproximación aplicada a la resolución de este problema es muy similar al del problema del viajante. Al tratarse de una imagen en escala de grises cada píxel toma un valor entre 0 y 255, de manera que para que se dé una continuidad dentro de la imagen (que las cada fila sea colocado al lado de su adyacente original) se debe reducir la distancia de valor que existe entre cada una de las distintas filas. Esta reducción por tanto se convierte en la función de fitness que se aplicará en este problema.

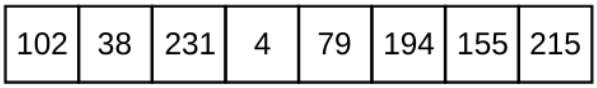


Figura 2: Vector de Alelos, cada uno representa una fila en la imagen.

La función fitness aplicada en este ejercicio se obtiene sumando el cuadrado de las diferencias existentes entre cada uno de los píxeles de las filas. De esta forma las diferencias siempre son positivas y se obtiene un buen parámetro de fitness que aplicar en cada iteración. En cuanto a la población se trata de un conjunto de individuos con un único gen de longitud equivalente a la longitud de la imagen a lo largo del eje y, de forma que cada alelo se corresponde con una fila de píxeles. En cada iteración cada uno de los individuos sufre una mutación por la cual un alelo o un conjunto de alelos son desplazados de forma aleatoria en bloque a otro punto del gen (insert), a continuación se calcula el fitness obtenido tras la mutación de dichos individuos y aquél que haya alcanzado una mayor reducción del error de la imagen es utilizado como progenitor para la creación de

una nueva generación. El tamaño del vector de alelos seleccionado para sufrir mutación se ve modificado a medida que avanza el algoritmo, aumentando de forma exponencial hasta un límite prefijado, siguiendo la siguiente regla, donde T indica el tamaño del vector e it la iteración correspondiente.

$$T_{vector} = \frac{1}{1 - (it_{Actual}/it_{Total})}$$

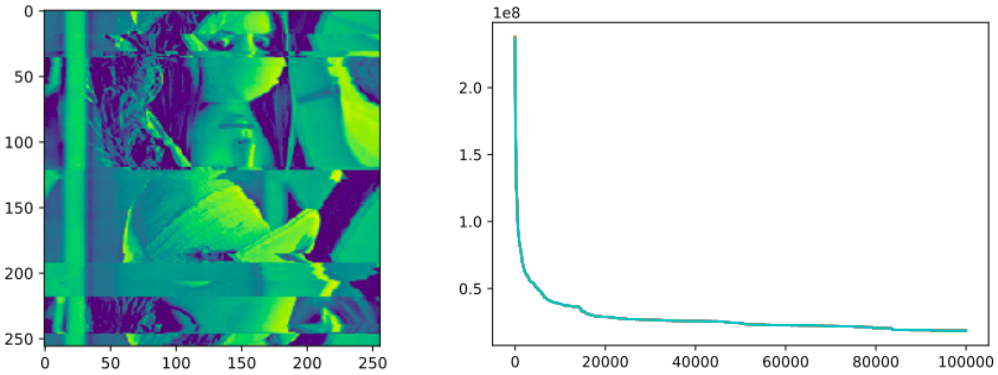
Es una forma de *Simulated Annealing* que garantiza que los cambios sufridos sean más precisos al inicio, donde la mayoría de los alelos se encuentran desligados de sus adyacentes originales, y más grandes hacia el final donde las filas de la imagen reconstruida han comenzado a formar bloques de dimensión mayor.

De igual forma en cada iteración dos individuos seleccionados de manera aleatoria sufren cruce controlado por el cual los genes de los individuos resultantes siempre codifican una solución posible (es decir, no existen dos alelos iguales y cada uno de ellos sigue correspondiendo a una única fila). El tipo de cruce empleado es el *Order Crossover Operator* en el cual se obtiene una nueva generación eligiendo una subcadena de uno de los padres y preservando el orden relativo de los elementos del otro padre. Por ejemplo, si se consideran las siguientes dos cadenas principales: 1423 y 4321 y tomamos el punto de corte como la mitad del gen tendríamos que los individuos resultantes obtienen como inicio la cadena 14** y 43** correspondiente a cada padre. A continuación se completan los alelos restantes en el orden relativo al otro padre, por lo que las cadenas finales serían: 1432 y 4312.

3. Resultados

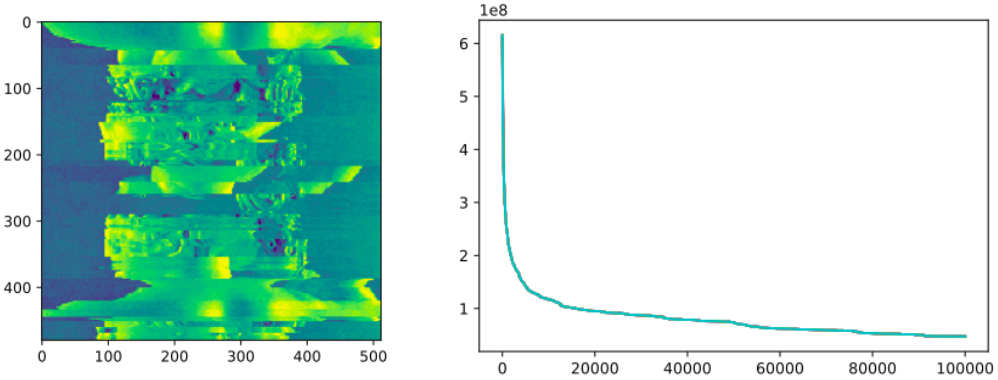
Los episodios para cada una de las imágenes han sido realizados con una población de 10 individuos y 100.000 iteraciones y los resultados obtenidos han sido los siguientes para cada una de las imágenes, representados en las figuras 3, 4 y 5.

Cada una de las distintas imágenes tiene una resolución cada vez mayor, de forma que la complejidad aumenta exponencialmente y a igualdad de individuos e iteraciones la decodificación tarda más en procesarse y alcanza un estado de resolución mucho menos avanzado. Esto se ve reflejado en la gráfica adyacente donde el error de la función de fitness es mucho mayor cuanto mayor es la resolución de la imagen.



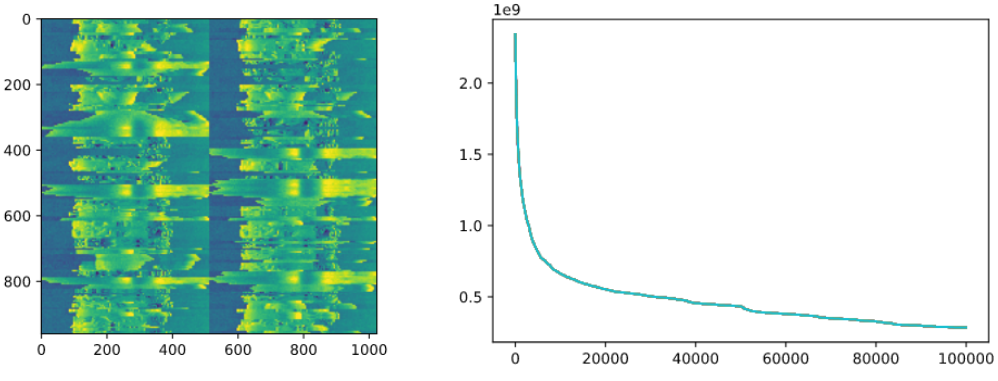
(a) Imagen Reconstruida (b) Error vs Iteración

Figura 3: Decodificación Imagen 256x256



(a) Imagen Reconstruida (b) Error vs Iteración

Figura 4: Decodificación Imagen 480x512



(a) Imagen Reconstruida (b) Error vs Iteración

Figura 5: Decodificación Imagen 960x1024

Como se aprecia en las gráficas de las imágenes 3, 4 y 5 al inicio la reducción es muy pronunciada ya que cualquier mínimo cambio sufrido conlleva con una alta probabilidad un descenso en la entropía del problema, sin embargo, a medida que los distintos individuos convergen hacia la solución óptima, la cual se desconoce inicialmente, la posibilidad de realizar una mutación que avance en el fitness se reduce. Los tiempos de cómputo han sido de 2 horas 36 minutos, 5 horas 24 minutos y 10 horas 50 minutos para cada una de las imágenes a decodificar.

4. Análisis de los Resultados

Observamos que el planteamiento descrito en el punto 2 es efectivo aunque no completamente eficaz en el ratio resultado/rendimiento. En primer lugar los tiempos de computación son extremadamente elevados lo cual debería llevar a reexaminar cada uno de los puntos mencionados (cruce, mutación y función de fitness).

Concretamente, contrario al proceso habitual en el cual la mutación se realiza en un porcentaje reducido de la población, se ha aplicado un cambio en todos los individuos y en todas las iteraciones. Por otro lado sucede lo opuesto en el caso del cruce, el cual debe realizarse con mayor frecuencia y que en este caso tan sólo se ha aplicado a dos individuos al azar en cada una de las iteraciones. En cuanto a la función de fitness descrita en el apartado segundo podría decirse que "pierde" información en su cálculo: la distancia entre los píxeles de cada fila se ignora ya que se realiza un cómputo del total de distancias, de manera que una forma más directa que se podría plantear para la función de fitness sería a través de un vector de distancias y no a través de un único entero. De esta forma se podría guiar a cada individuo a explorar mutaciones entre alelos contiguos cuya aportación al error total sea elevada, es decir, entre filas muy dispares entre sí.

En segundo lugar se ha introducido un hiperparámetro dentro del algoritmo: el número de alelos que sufren mutación en cada iteración. Este hiperparámetro podría ser eliminado y sustituido por un número aleatorio. En tercer lugar, las imágenes resultantes están inconclusas aunque se puede apreciar perfectamente en la imagen 3 una chica y un sombrero, mientras que en las imágenes 4 y 5 sería necesario ampliar el número de individuos o el número de iteraciones para poder clarificar mejor la información contenida en ellas. En cualquier caso se observa que la imagen 5 es una repetición de la imagen 4.

5. Conclusiones

Este ejercicio es un ejemplo muy concreto de cómo aplicar algoritmos evolutivos a la resolución de problemas. En este caso concreto se sabe que existe una solución óptima (la

imagen original), y cuya verificación al ojo humano se realiza de forma casi instantánea, sin embargo no es el caso general, ya que la mayoría de las veces se desconoce si la solución alcanzada coincide con el mínimo global o con un mínimo local.

Toda la documentación y el código empleado pueden ser consultados en el siguiente enlace que conduce al repositorio.