

Tarea Final Algoritmos bioinspirados y Técnicas de Computacion Evolutiva: Problema de los reproductores mediante genético

Elias Barba Moral

Algoritmos bioinspirados y Técnicas de Computacion Evolutiva

April 4, 2017

1 Planteamiento del problema

Enunciado del problema: En producción animal es necesario controlar la consanguinidad de los futuros reproductores. Esta matriz indica la relación de parentesco entre parejas de 215 individuos de una generación. Se pretende seleccionar 3 lotes de 30 individuos cada uno, de forma que, en primer lugar, el máximo de las relaciones en los lotes sea lo menor posible y, en segundo lugar, que la media de las relaciones en los lotes sea lo menor posible.

En este problema hemos de seleccionar los 90 individuos y separarlos en 3 grupos iguales que tienen menor relación de parentesco con fines reproductivos. Al ser la cantidad total de posibles combinaciones enorme y el coste de evaluarlas todas desmedido, la aplicación de un método heurístico parece el método mas recomendable para resolver el problema. Concretamente usaremos un algoritmo de tipo genético.

Los algoritmos genéticos son un tipo de algoritmos que se basan en la idea de la selección natural de las mejores cadenas genéticas. Los cambios en las soluciones se producen básicamente debido a dos factores: la mutación y el cruce del material genético de los padres.

La mutación es un proceso por el cual el material genético de un individuo cambia espontáneamente. Estos cambios pueden otorgar a dicho individuo una serie de beneficios que lo conviertan en mas resistente a las amenazas en su entorno. Si el individuo es mas fuerte, aumentarán sus posibilidades de reproducirse y pasar el material genético mutado que le ha dado ventaja a sus descendientes.

El cruce de material genético de los padres es la otra forma de alterar el material genético. En general, en los procesos biológicos el material genético de los hijos se compone de la mezcla del material genético de dos padres a partes iguales.

Ambas técnicas serán incluidas en la creación de un algoritmo para resolver el problema presentado.

2 Desarrollo del Primer algoritmo genético

Para el estudio del problema creé una lista en R (en el código tiene el nombre "individuo"), que está compuesta por tres elementos (uno para cada lote). En cada elemento de dicha lista hay 30 numeros, elegidos inicialmente al azar entre 1 y 215, sin reemplazamiento. Posteriormente creé un vector con los numeros que no estaban incluidos en ninguno de los tres lotes, con nombre "poolofuntakenfish".

En la función principal, creé un bucle con un numero grande de iteraciones, donde en cada iteracion el individuo pasará por un proceso de mejora tanto por reproducción como de mutación. La reproducción será asexual, un individuo será capaz de producir un hijo por si mismo sin necesidad de otro individuo.

En esta reproducción se seleccionan los dos peores lotes del individuo y se seleccionan las posiciones aleatorias de 10 genes. Los peces que ocupan esos genes se intercambian de lote y se obtiene un nuevo individuo, el hijo. Si la máxima relación de los peces en el hijo es mas pequeña que en el padre, entonces se sustituye al padre por el hijo. En caso contrario, el algoritmo sigue adelante, quedándose como individuo al padre.

La mutación consiste en la mejora del individuo sustituyendo los peces que crean el máximo de relación en cada uno de los lotes. Estos peces se sustituyen por otros tomados aleatoriamente de la "pool of unfittest fish". Si la mutación mejora el máximo del lote se guarda y se actualiza al individuo.

El código tiene en el bucle un apartado para ir dibujando los resultados y seguir el progreso del algoritmo. Además al final aparece una descripción del mejor individuo encontrado. Recordar que este individuo no sólo se guarda si la máxima relación es mas pequeña, si no que en caso que el máximo sea igual, busca el que individuo con una media mas pequeña. Además de las funciones descritas en el código aparecen otras funciones similares, que se usan en el desarrollo del código. Estas funciones son prescindibles, pero se podría estudiar cual de todas mejora de forma más rápida.

3 Desarrollo del Segundo algoritmo genético

La idea de este segundo algoritmo genético es similar a la del anterior algoritmo, pero en este caso aplicándolo a una población, no a un sólo individuo. Para este caso creé una lista en el que cada elemento era un individuo, y venía descrito por otra lista con los tres lotes y los peces no incluidos en ninguno de los otros lotes.

En el bucle principal selecciono al peor individuo (con un máximo mas grande) de la población, y le aplico un número de iteraciones ("nIter") para mejorar dicho individuo. Ese numero de iteraciones va creciendo a medida que el algoritmo progresa. En el caso que no se encuentre una distribución que mejore la anterior, se pone en marcha un contador y una vez llegado a un limite ("límite contador"), si no se ha encontrado ninguna mejora, dicho individuo se descarta, y se crea uno nuevo.

El método de mejora del individuo es el mismo que en el algoritmo anterior: primero la misma fase de reproducción asexual y a continuación la misma fase de mutación. Esto puede ayudar a la hora de comparar ambos algoritmos. De la misma forma el algoritmo cuenta con una parte para realizar una representación grafica de la evolución de las soluciones y una representación final de la mejor solución.

4 Resultados

En primer lugar analizaremos los resultados que devuelve el primer algoritmo. En dos ejecuciones de 20000 iteraciones (aproximadamente 160 minutos) arrojan las siguientes representaciones gráficas:

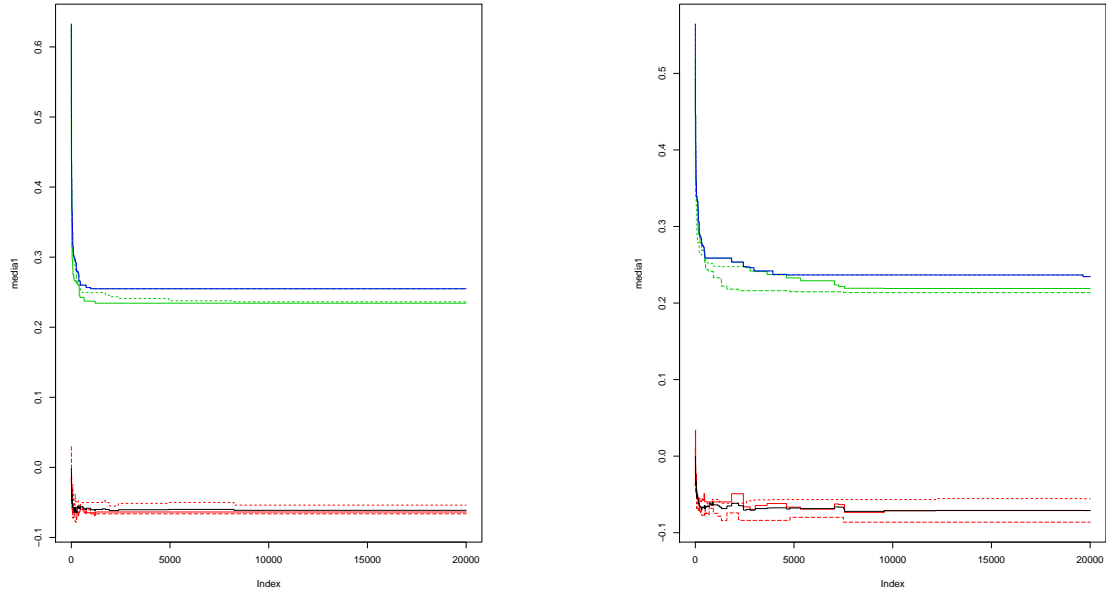


Figure 1: En las figuras la línea azul representa el máximo relatedness del individuo, mientras que las líneas verdes representan el máximo relatedness de los otros dos lotes que no son máximos. La línea negra representa la media de las líneas rojas, que son las medias de relatedness de cada lote.

Se puede apreciar un descenso pronunciado al principio del algoritmo, mejorando mucho el individuo, para posteriormente estancarse en un determinado valor y prácticamente no mejorar.

```

Resultados finales:
Tiempo total para 20000 iteraciones: 7611.63 1633.48 9809.22 NA NA
Maximo del lote 1 : 0.2549
Maximo del lote 2 : 0.2363
Maximo del lote 3 : 0.2343
Media del maximo de los tres lotes: 0.2418333
Media de la media de los tres lotes: -0.0612587

Peces que forman los lotes:
El lote 1 lo componen:
148 40 104 178 146 137 81 174 82 58 28 31 8 6 201 185 43 32 85 213 149
22 177 20 165 50 7 210 143 122
El lote 2 lo componen:
101 180 203 140 120 173 184 91 87 3 110 196 169 103 98 186 27 160 106 13
3 207 167 2 208 156 38 74 194 23 62
El lote 3 lo componen:
13 200 164 72 29 111 198 128 46 4 59 114 49 95 127 102 93 141 90 195 15
0 14 105 192 135 34 39 84 154 52

Resultados finales:
Tiempo total para 20000 iteraciones: 7522.56 1633.72 9648.38 NA NA
Maximo del lote 1 : 0.2136
Maximo del lote 2 : 0.2345
Maximo del lote 3 : 0.219
Media del maximo de los tres lotes: 0.2223667
Media de la media de los tres lotes: -0.07105073

Peces que forman los lotes:
El lote 1 lo componen:
46 65 66 91 127 81 24 60 44 187 93 128 185 62 82 34 114 50 15 171 174
31 135 195 206 90 178 96 173 161
El lote 2 lo componen:
27 180 164 129 8 45 32 156 188 160 7 115 186 64 97 208 179 204 67 12 4
52 118 111 150 104 13 108 72 198
El lote 3 lo componen:
9 197 2 28 165 143 200 77 43 57 10 19 109 190 23 137 172 49 16 145 53
210 35 25 152 120 116 3 105 110

```

Figure 2: Resultados del primer algoritmo como son devueltos a la ejecución del código en R.

Vemos que los mejores individuos guardan poca relación entre sí. En cuanto al segundo algoritmo, dos ejecuciones de 1000 y 2000 (110 y 220 minutos aproximadamente) iteraciones devuelven la siguiente representación gráfica:

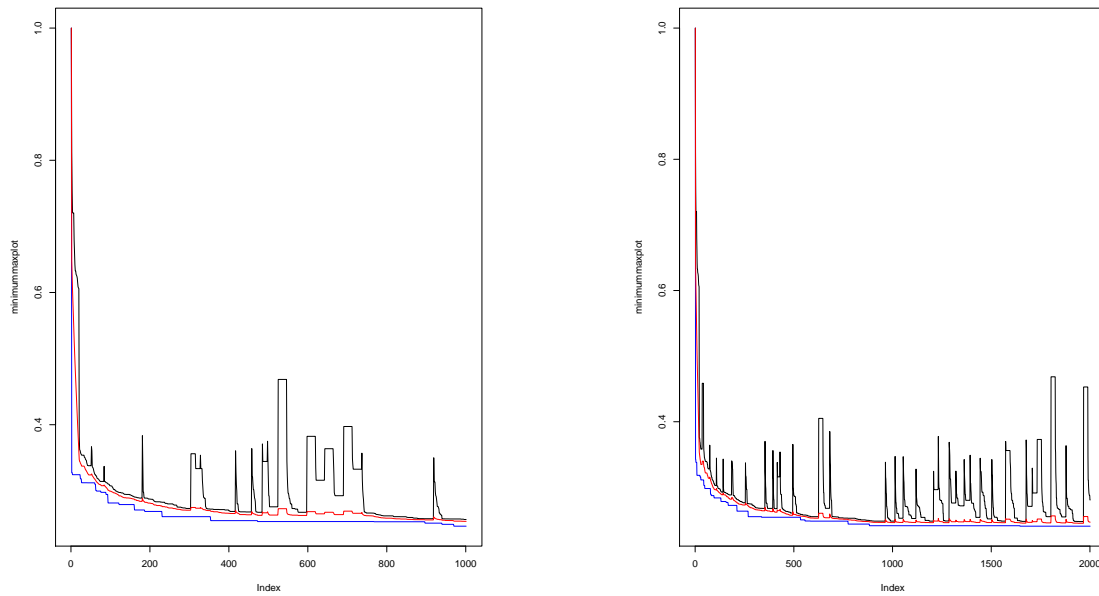


Figure 3: En las figuras la línea azul representa el máximo relatedness del individuo con dicho valor mas bajo. La línea negra el el máximo relatedness del individuo con dicho valor mas alto, y la línea roja representa la media de los máximos de todos los individuos.

Similarmente al algoritmo anterior se ve una mejora clara, aunque menos pronunciada. Posteriormente el algoritmo es capaz de seguir mejorandose a un ritmo cada vez mas lento. Los picos negros son puntos en los que un individuo queda estancado y se crea una nueva solución, que se mejora consecutivamente hasta llegar a los valores del resto de individuos. Vemos que la cantidad de individuos estancados aumenta con el numero de iteraciones ya que para mejorar cada individuo es necesario mas iteraciones de la parte de mejora. Las soluciones finales que se aportan son:

```

Resultados finales:
Maximo del lote 1 : 0.2389
Maximo del lote 2 : 0.2174
Maximo del lote 3 : 0.2465
Media de la media de los tres lotes: -0.06423318

Peces que forman los lotes:
El lote 1 lo componen:
108 8 113 124 65 62 161 93 19 84 76 116 135 28 136 125 183 180 105 4 59
205 195 15 24 194 2 29 211 91
El lote 2 lo componen:
165 128 97 90 16 73 96 208 127 46 50 111 114 54 52 143 110 118 97 34 18
6 197 156 209 100 153 72 7 149 200
El lote 3 lo componen:
10 83 150 170 117 6 144 104 109 27 115 32 191 174 25 23 120 206 137 210
187 126 146 185 141 82 12 173 119 43

Resultados finales:
Tiempo total para 2000 iteraciones: 12938.83 43.54 13271.62 NA NA
Maximo del lote 1 : 0.2408
Maximo del lote 2 : 0.2273
Maximo del lote 3 : 0.2242
Media de la media de los tres lotes: -0.05845693

Peces que forman los lotes:
El lote 1 lo componen:
109 34 7 186 127 105 113 43 59 190 72 206 50 204 49 197 80 10 39 108 10
3 106 169 143 93 210 149 66 170 174
El lote 2 lo componen:
64 136 114 146 118 4 32 87 164 139 101 12 31 177 44 27 191 17 120 137 9
23 97 150 86 28 91 192 75 171
El lote 3 lo componen:
52 95 173 152 15 19 156 165 122 81 153 24 8 46 180 111 84 200 194 85 16
124 145 62 141 161 29 110 185 119

```

Figure 4: Resultados del segundo algoritmo como son devueltos a la ejecución del código en R.

Del mismo modo que en el anterior algoritmo los lotes de los mejores individuos tienen poca relación en ambos casos.

5 Conclusiones

Podemos concluir afirmando que el primer algoritmo hace un buen trabajo de aproximación a una buena solución. En caso de querer una buena solución de forma rápida este es el mejor método. Sin embargo este método se estanca en una región con mucha facilidad y se deja sin explorar gran parte de las soluciones posibles.

El segundo algoritmo deja menos regiones sin explorar, y por tanto, a largo plazo, es capaz de encontrar mejores soluciones que el primero. Sin embargo es un algoritmo mucho más lento y que necesita cada vez más tiempo para encontrar una solución mejor.

Dado que este es un método heurístico y la solución que devolvamos sabemos que no tiene por qué ser la óptima, nos interesa saber que cerca nos quedamos de la mejor solución. En este caso, viendo el comportamiento exponencial en general del segundo método es posible ajustar una exponencial y estimar en el infinito a que valor va a tender el algoritmo.

Otra posible mejora sería estudiar la conveniencia de la función de reproducción a partir de un determinado número de iteraciones. Dado que es un cambio bastante grande en la composición del individuo es posible que en la mayoría de casos se deseche la nueva solución, gastando tiempo de computación que se podría usar en buscar otro método menos agresivo.