Algoritmos genéticos: operadores

LAURA DEL PINO DÍAZ – GRADO EN INGENIERÍA INFORMÁTICA – MENCIÓN EN COMPUTACIÓN

SISTEMAS INTELIGENTES 1

índice

|  |  |
| --- | --- |
| Objetivo | 2 |
| La codificación del problema | **2** |
| La función de fitness | **3** |
| Función de terminación del algoritmo | **3** |
| Operador de selección | **3** |
| Operador de cruce | **4** |
| Operador de mutación | **5** |
| Valores de los parámetros ajustados | **6** |
| Conclusión | **8** |

oBJETIVO

Los objetivos de esta práctica son:

* Montar la estructura de datos de la red de tráfico y el simulador basado en autómatas celulares.
* Diseñar la codificación apropiada para el problema a resolver.
* Diseñar la funcióin de fitness basada en la maximización del número de vehículos por minuto que salen de la red durante la simulación.
* Diseñar los operadores genéticos a utilizar y los parámetros: tamaño de la población, probabilidad de mutación, número de generaciones.
* Programar el algoritmo genético.
* Ejecutarlo e iterat al punto 3 hasta que se obtengan los mejores resultados que nos de confianza de encontrar el óptimo en la mayoría de las ejecuciones del algoritmo, evitando la convergencia prematura.
* Representar la evolución del fitness máximo y medio de la población.
* Opcionalmente representar visualmente las simulaciones, en particular las que se produzcan con los mejores resultados del ciclo de los semáforos.

La codificación del problema

Para aplicar las distintas operaciones propias de los algoritmos genéticos, se decide utilizar la codificación binaria de cuatro de los semáforos ya que se pueden emparejar de dos en dos de forma que para el valor de uno el otro tiene el valor opuesto.

Se representan doce estados que pueden tener los semáforos durante los dos minutos virtuales de la simulación que se repetirán hasta llegar a las dos horas de la simulación al completo. De forma que el cromosoma de un individuo queda como una matriz de 12x4 1’s y 0’s. Además el conjunto de todos los cromosomas individuales se pueden ver como una matriz de 12x4xN siendo N el tamaño de la población.

LA función de fitness

Para determinar la bondad de un cromosoma o una posible solución se utiliza la función de fitness, en este caso hemos utilizado el conteo del número de coches que han salido de la simulación con respecto del número que ha entrado al final de la simulación, de esta forma todos los valores que devuelva el simulator caerán dentro del intervalo .

Cabe destacar que el simulador se ha desarrollado en Java, de forma que facilitase el modelado del simulador. Mientras que para el desarrollo del algoritmo genético mantenemos MatLab como lenguaje de desarrollo por su facilidad en la gestión de las matrices.

Función de terminación

Para esta práctica se ha optado que la condición de terminación del algoritmo sea un número fijo de iteraciones/generaciones que se pueda pasar como parámetro y, por tanto, sea controlable por nosotros desde la línea de comandos de MatLab.

operador de selección

El operador de selección de este algoritmo genético es el denominado “selección por torneo probabilístico” que consiste en seleccionar dos cromosomas del conjunto de cromosomas y ordenarlos de mayor a menor fitness. Tras ello, se elige un número aleatorio entre 0 y 1, si éste número es mayor que una probabilidad que nos viene dada por parámetro se elige al cromosoma seleccionado con mayor fitness, en caso contrario al que tiene peor fitness.

La probabilidad viene condicionada también por el número de iteraciones del algoritmo de forma que la probabilidad que usará el algoritmo es el que determinó el usuario desde la línea de comandos partido el número de iteraciones del algoritmo, de esta forma el operador evoluciona hasta quedarse en la mayoría de las ocasiones con los mejores cromosomas en las últimas iteraciones.

Este operador nos permite da una oportunidad a aquellos individuos que tienen peor fitness pero pueden generar mejores soluciones que las que tenemos en esta generación. Además, permite que aquellos con un buen fitness se reproduzcan más veces.

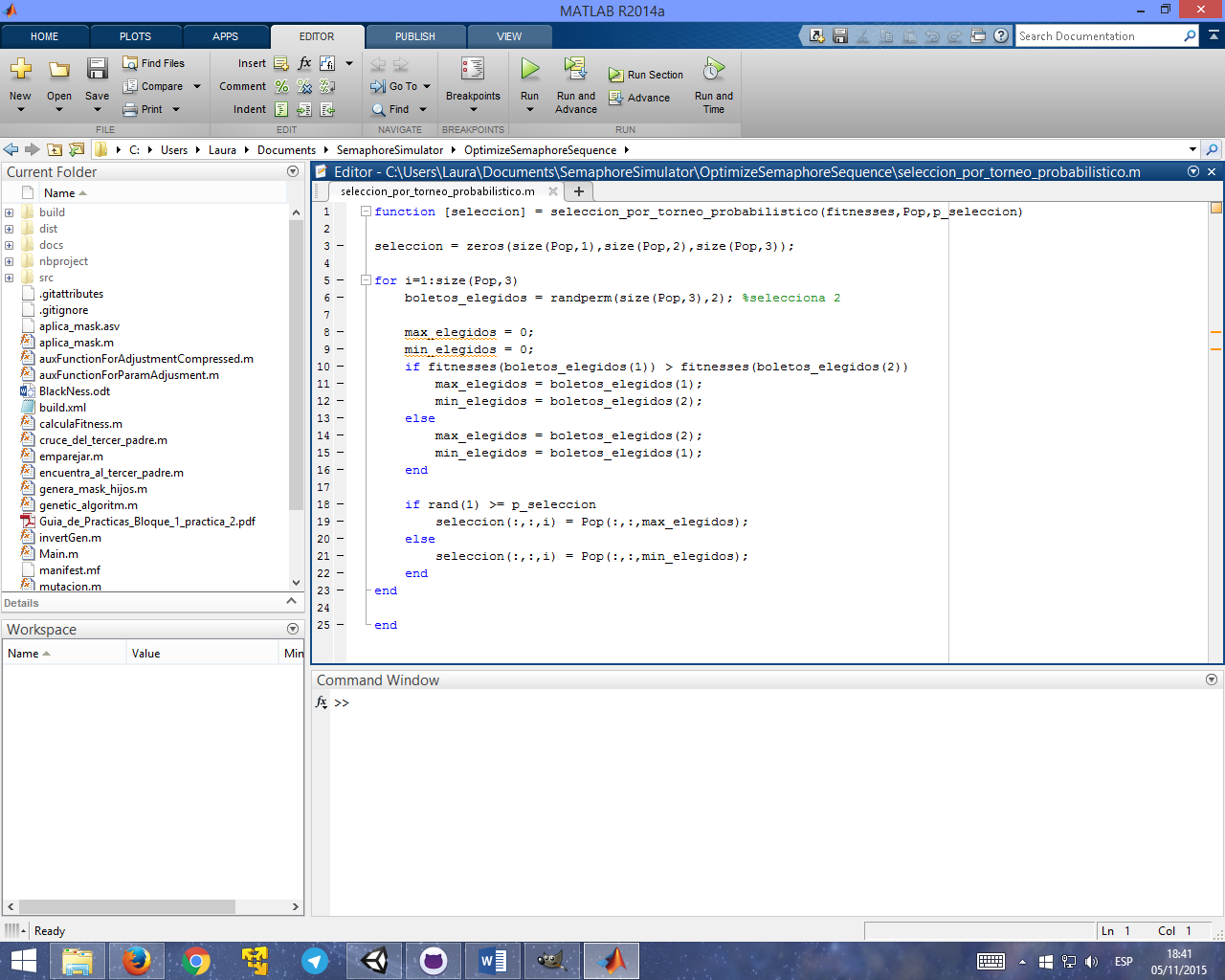


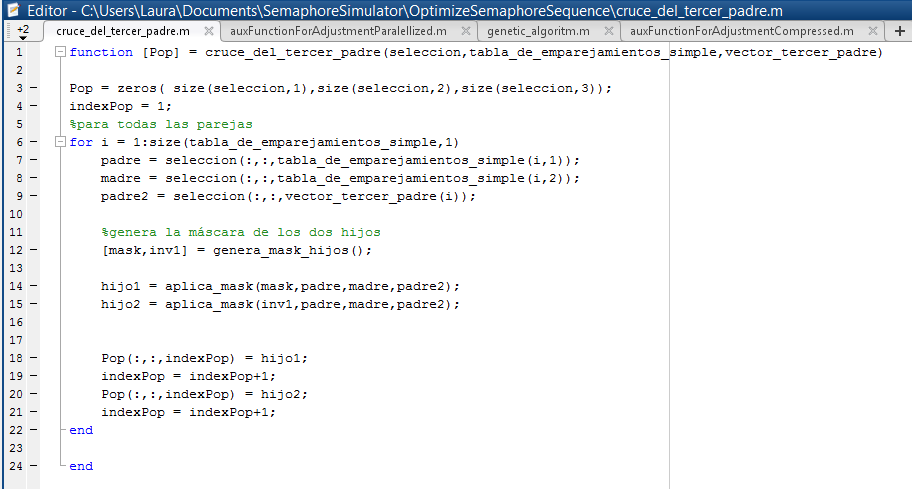
Ilustración 1: Operador de selección "por torneo probabilístico".

operador de cruce

El operador de cruce que se ha seleccionado para esta práctica es el denominado “cruce del tercer padre”. Consiste en dado tres padres, se genera una máscara del mismo tamaño que el cromosoma original pero contiene los números desde 0 hasta 2 de forma que indica de qué padre tomará el hijo el gen correspondiente.

Para generar los posibles hermanos de la primera solución se pueden generar las “inversas” de la máscara, para ello se suma uno a la máscara del hermano anterior en todas las posiciones y se le aplica la operación del módulo.

En esta implementación se usa como número de padres 3 y número de hijos generados 2,puesto que el tercer padre lo hemos elegido de la siguiente pareja a cruzar, esto se hace con el objetivo de dar mayor diversidad genética y a la vez mantener el mismo número de individuos en cada iteración.



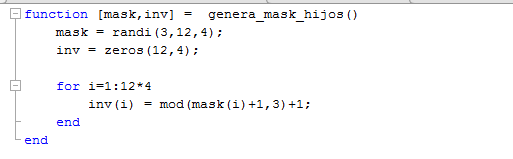


Ilustración 2: Código del cruce del tercer padre y la generación de la máscara de los hijos

operador de mutación

El operador mutación se basa en que todos los cromosomas de la nueva generación tienen la misma probabilidad de mutar uno de sus genes, elegido arbitrariamente en el momento que se determina que ese cromosoma muta, pero a diferencia que la práctica anterior la probabilidad de mutar decrementa con el número de generaciones, de forma que al principio es muy fácil que un individuo mute pero al final es muy poco probable.

algoritmo genético completo

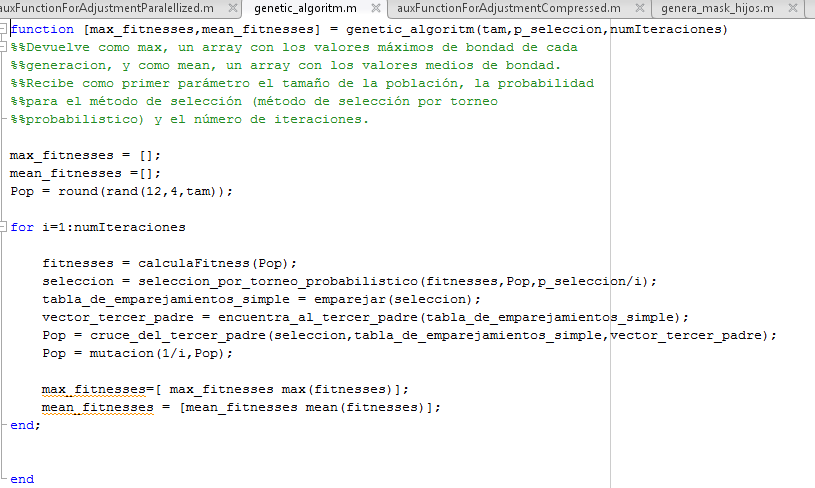


Ilustración 3: Cuerpo del algoritmo genético

valor de los parámetros ajustados

A continuación se adjunta una lista de los resultados obtenidos para los distintos parámetros que se piden ajustar :

conclusión

En esta práctica hemos aprendido:



bibliografía

* Genetic algoritm with 3-parent uniform crossover – V.Edmondson & B. Gillett

<http://cs.mst.edu/media/academic/cs/documents/technicalreports/93-22.pdf> (Página 17 de 180)