

SISTEMAS INTELIGENTES I

Bloque 1: Algoritmos Genéticos

Guía de Prácticas



Javier J. Sánchez Medina
javier.sanchez@ulpgc.es

El propósito de esta guía de prácticas es dotar a los/as alumnos/as de Sistemas Inteligentes I, de los rudimentos prácticos en diseño y programación de Algoritmos Genéticos, Redes Neuronales y una introducción a la Minería de Datos

Esto complementa los contenidos teóricos que de esta asignatura serán impartidos.

La asignatura consta de 15 horas de prácticas de Aula y 15 horas de prácticas de Laboratorio. Se dividirán en los tres bloques arriba mencionados:

Bloque 1: Algoritmos Genéticos (6 semanas)
Bloque 2: Redes Neuronales Artificiales (6 semanas)
Bloque 3: Minería de Datos (2 semanas)

Las prácticas son individuales.

Práctica 3: Algoritmos Genéticos. Optimización Multicriterio. Frontera de Pareto.

Introducción:

La mayoría de los problemas reales de optimización en ingeniería no son problemas en los cuales se pueda resolver optimizando una única función de *fitness*. Por lo general, se deben compaginar varios criterios de optimización, como peso vs. precio, velocidad vs. calidad, fiabilidad vs. prestación punta, etc.

El propósito de esta práctica es trabajar con más de un criterio de optimización. En esos casos surge un concepto clásico en optimización no determinística (no sólo en Algoritmos Genéticos) como es la llamada “Frontera de Pareto”. Cuando se trata de optimizar una o más funciones de *fitness*, es preciso primero evaluar como combinar los valores de estas N funciones, y luego determinar cómo obtener el conjunto de posibles soluciones óptimas, en las cuales se alcanza ese valor máximo (o mínimo, si se trata de una minimización), en la combinación de ambas.

Hay muchas formas de obtener ese espacio de soluciones. La mayoría de las técnicas estándar para ello caen dentro del conjunto denominado técnicas “a posteriori”. Las más recurridas en la bibliografía son las siguientes:

- Normal Boundary Intersection (NBI)
- Modified Normal Boundary Intersection (NBIm)
- Normal Constraint (NC)
- Successive Pareto Optimization (SPO)
- Directed Search Domain (DSD)
- Non-dominated Sorting Genetic Algorithm-II (NSGA-II)
- Strength Pareto Evolutionary Algorithm 2 (SPEA-2)

Las dos últimas técnicas son del tipo de computación evolutiva, muy aplicables a los algoritmos genéticos.

El propósito de esta práctica es resolver un problema de optimización multicriterio, obteniendo la frontera de Pareto, mediante múltiples iteraciones del GA, que es la forma más sencilla de obtenerla.

Para ello, ajustaremos el algoritmo genético para obtener el mejor *fitness* posible y luego ejecutaremos dicho algoritmo tantas veces como sea necesario para representar la frontera de Pareto, con los mejores valores de *fitness* solamente.

Problema a resolver:

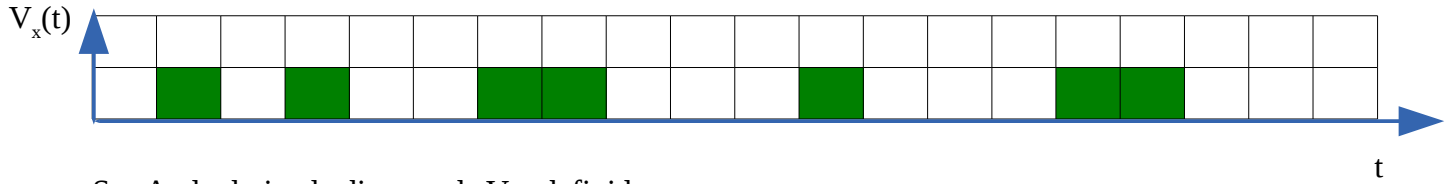
Vamos a extender el problema de la práctica 2 para convertirlo en un problema de optimización multicriterio (de dos criterios). En la práctica 2 se sugirió tomar el ancho de banda de salida como función de *fitness*. Ahora vamos a combinar esa variable con otra estrechamente relacionada con las emisiones, suponiendo que los vehículos emitan gases con efecto invernadero (GHGs).

Se sabe que existe una relación entre las variaciones de velocidad del tráfico y el nivel de emisiones de GHGs. Un tráfico con pocas variaciones de velocidad, pocas ondas de “stop-and-go” emite menos nivel de gases.

Con esta idea, vamos a definir un criterio de optimización que se base en la minimización del área

de la señal de aceleración de los vehículos.

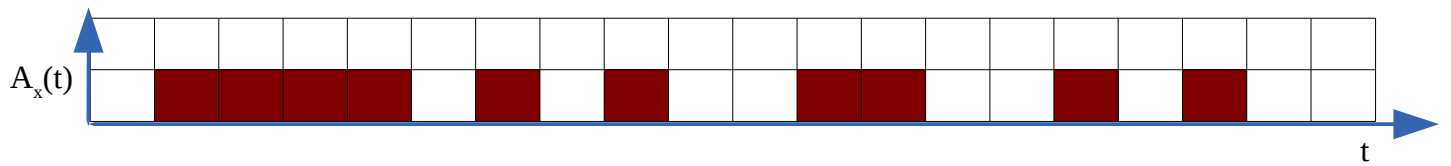
Sea V_x la señal de velocidad de un vehículo en nuestra simulación. Sus valores alternarán entre 0 y 1 según se mueva o no en cada iteración del simulador, por ejemplo, como sigue:



Sea A_x la derivada discreta de V_x , definida como:

$$A_x(t) = \frac{V_x(t) - V_x(t-1)}{t - (t-1)}$$

La señal de A_x (en valor absoluto) para la anterior señal de V_x será tal que así (asumiendo V_x inicial = 0):



El área total de la señal A_x nos da una pista del nivel de emisiones que ha podido producir ese vehículo.

Debemos monitorizar el estado de todos los vehículos simulados durante 2 horas (7200 iteraciones/segundos de simulación) y calcular el Área de la señal de aceleración de todos ellos y la media de este valor.

Una vez obtenido ese valor medio, la componente de este criterio en la función de *fitness* será:

$$f_e(ind) = 1 - \left(\frac{AVG(AREA(a))}{N_{iteraciones}} \right), \forall a \in \{a_1, a_2, a_3, \dots, a_{N_{veh}}\}$$

La otra componente de la función de *fitness* es la que comentábamos anteriormente, relativa al ancho de banda total de vehículos que abandonan la red. Se puede definir como sigue:

$$f_{th}(ind) = \frac{N_{veh}}{N_{iteraciones}}$$

Por último, vamos a definir una función de optimización multicriterio lineal tal cual sigue:

$$f(ind) = 0,5 \times f_{th} + 0,5 \times f_e$$

Objetivos de la práctica:

Los objetivos de esta práctica se pueden resumir como sigue:

0: Modificar el simulador y algoritmo genético de la práctica anterior para que incluya una estructura de datos en la cual se almacene el área de la aceleración de cada vehículo que circula por la red.

1: Ajustar el algoritmo genético para lograr lo más consistentemente posible un valor de *fitness* lo mejor posible.

2: Ejecutar el algoritmo genético un número de veces que permita la visualización de la frontera de Pareto de este ejercicio, representando un número suficiente de pares de valores de los dos criterios de optimización, de los mejores resultados de *fitness*.

Entregables:

Memoria de menos de 10 páginas con un anexo con el código Matlab (u otros) con el algoritmo genético y el simulador de tráfico basado en autómatas celulares..

La memoria debe incluir:

- Código nuevo añadido a la práctica 2, en particular, para el *tracking* de las velocidades de los vehículos, y el cálculo del área de sus aceleraciones
- Operadores utilizado de selección, cruce, mutación y condición de finalización
- Valores de los parámetros ajustados:
 - Tamaño de la población
 - Número de generaciones
 - Probabilidad de Mutación
- Representación de la evolución del *fitness* máximo y medio para los mejores individuos
- Representación de la frontera de Pareto
- Opcionalmente, interpolar una curva que se ajuste lo mejor posible a la frontera de Pareto, utilizando por ejemplo la función “Polyfit” de Matlab. Representar la curva interpolada sobre la nube de puntos de la frontera de Pareto.