

EVANNAI

A. Baldominos, P. Isasi & Y. Sáez

Algoritmos Genéticos y Evolutivos

Grado en Ingeniería Informática

Curso 2016-2017

# Práctica 1

## Optimización de Sensores en Smart Cities





# Motivación

- En 2008, el 50% de la población vivía en ciudades
- Hay 21 megaciudades (+10 millones de habitantes)
- ¡Tokyo tiene +36 millones de habitantes!
- Las ciudades consumen el 60-80% de la energía medida anualmente
- Las *smart cities* podrían mejorar la calidad de vida y el impacto medioambiental

Fuente: <http://www.postscapes.com/anatomy-of-a-smart-city/>



# Contexto

- La ciudad de Madrid cuenta con 24 estaciones de control de calidad del aire
- El portal de Open Data del Ayto. de Madrid publica:
  - Datos horarios y diarios de calidad del aire (2001-2016)
  - Datos en tiempo real de calidad del aire
  - Ubicación de las estaciones de control de calidad del aire
- La deuda de la ciudad en 2015 era de 4,7 M€

Fuente: <http://datos.madrid.es>





# Datos

- Las estaciones de control de calidad del aire miden:

1	Dióxido de Azufre	SO <sub>2</sub>	µg/m <sup>3</sup>
2	Monóxido de Carbono	CO	µg/m <sup>3</sup>
3	Monóxido de Nitrógeno	NO	µg/m <sup>3</sup>
4	Dióxido de Nitrógeno	NO <sub>2</sub>	µg/m <sup>3</sup>
5	Partículas < 2.5 µm	PM2.5	µg/m <sup>3</sup>
6	Partículas < 10 µm	PM10	µg/m <sup>3</sup>
7	Óxidos de Nitrógeno	NOx	µg/m <sup>3</sup>
8	Ozono	O <sub>3</sub>	µg/m <sup>3</sup>
9	Tolueno	TOL	µg/m <sup>3</sup>
10	Benceno	BEN	µg/m <sup>3</sup>
11	Etilbenceno	EBE	µg/m <sup>3</sup>
12	Metaxileno	MXY	µg/m <sup>3</sup>
13	Paraxileno	PXY	µg/m <sup>3</sup>
14	Ortoxileno	OXY	µg/m <sup>3</sup>
15	Hexano (total)	TCH	mg/m <sup>3</sup>
16	Hexano (no metánicos)	NMHC	mg/m <sup>3</sup>

Fuente: <http://datos.madrid.es>

# Objetivo

- Las estaciones tienen un coste, que depende de:
  - Coste de la base (fijo para todas las estaciones)
  - Coste por sensor
  - Coste de mantenimiento (variable según la ubicación)
- El Ayto. de Madrid necesita renovar las estaciones y al mismo tiempo busca reducir su deuda pública
- El Ayto. de Madrid pide reducir el número de sensores:
  - El coste sea mínimo
  - El sacrificio en precisión sea mínimo
- Vamos a emplear un **Algoritmo Genético**

# Codificación

- Cromosomas binarios
  - 0 -> el sensor no se instala en la estación correspondiente
  - 1 -> el sensor sí se instala
  - Si una estación no tiene ningún sensor, no se instala la base

- 384 genes:

- 24 estaciones x 16 sensores

- Ejemplo:

0100010101001010|0010010100000000|0000000000000000|...

Estación 1

CO, PM10, O<sub>3</sub>, BEN, PXY, TCH

Estación 2

NO, PM10, O<sub>3</sub>

Estación 3

No se instala

# Metodología

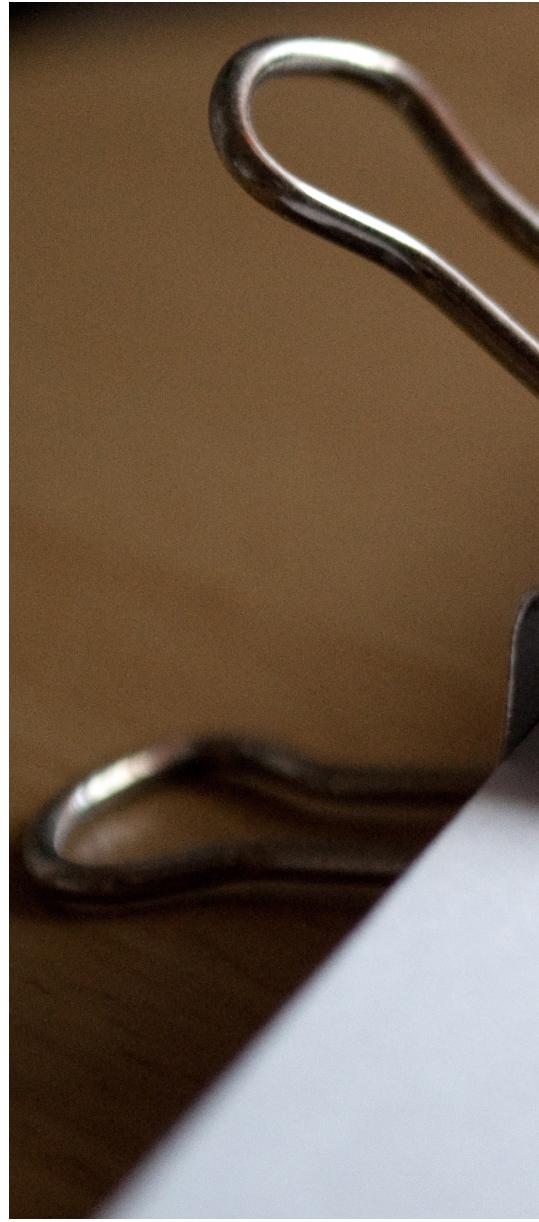
- Te proporcionamos un servidor que calcula automáticamente la función de *fitness*:

$$f = p/q$$

- $p$  es un valor proporcional al precio
- $q$  es un valor proporcional a la calidad (precisión)
- El objetivo es minimizar la función de fitness ( $\min=0$ )

# Simulador

- El simulador permite hacer pruebas
  - Entorno de pruebas reducido
  - Solo 4 estaciones -> cromosoma de 64 genes
- Durante el desarrollo se llamará al siguiente servicio web:  
`http://163.117.164.230/age?f=test&c=<chromosome>`
- Se realizará optimización mediante:
  - Fuerza bruta
  - Algoritmos Genéticos
  - (Opcional) Otras técnicas de computación biológica (ampliación)



# Entrega

- Memoria breve (10-20 páginas), describiendo:
  - Parámetros propuestos, y cuáles funcionan mejor
  - Análisis y comparativa de resultados **con gráficas y tablas**
  - Mejor resultado obtenido y **número de evaluaciones** necesarias (de media)
  - Problemas encontrados y conclusiones
- La práctica es individual
- Se debe subir a Aula Global (entregador) un fichero .zip con la memoria y el código fuente
- **Fecha prevista de entrega: 21 de octubre de 2016**

# Evaluación

- Se compararán las soluciones de los estudiantes
- Se pedirán pruebas adicionales, de mayor complejidad
- Se evaluarán:
  - Los mejores resultados obtenidos para cada prueba
  - El número de evaluaciones requerido para alcanzarlos
  - El tiempo de ejecución
  - El alumno debe mostrar estos datos en la salida del programa e incluirlos en la memoria
- También se evaluará la robustez del programa, ejecutándolo varias veces