

Algoritmos Genéticos en Python II: Introducción a los problemas con múltiples objetivos

Daniel Gutiérrez Reina, dgutierrezreina@us.es

SWYP: Students, Women in Engineering and Young Professionals,

Contenido día II

- ▶ Resumen del día I.
- ▶ Problemas con múltiples objetivos.
- ▶ Dominancia de Pareto.
- ▶ Algoritmos NSGA-II.
- ▶ El problema de la suma de subconjuntos.
- ▶ Ejemplos de problemas de Ingeniería.
- ▶ Por donde seguir



ACE-TI

Grupo de investigación
Ingeniería Electrónica

Motivación

¿Sabes qué es un algoritmo genético?

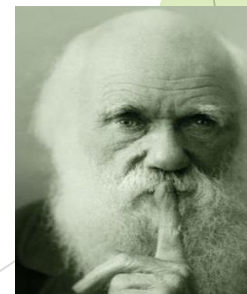
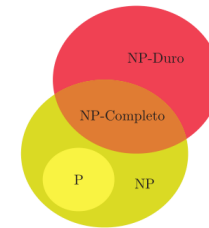
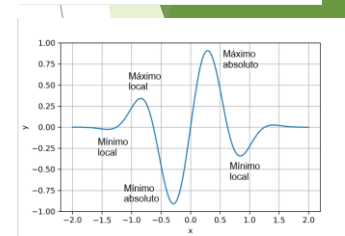
Un algoritmo de inteligencia artificial → Un herramienta

Un algoritmo de optimización → Maximizar / Minimizar.

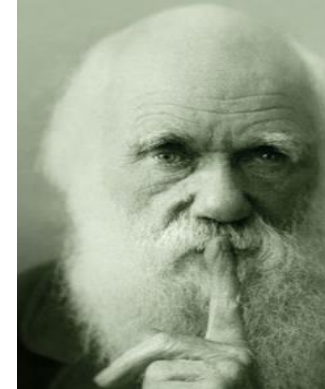
Resolver problemas específicos → Complejos.

Basado en la teoría de la evolución → Teoría de Darwin.

Idea de los años 70 → No es nada nuevo.



Teoría de la evolución de Darwin



ACE-TI

Grupo de investigación
Ingeniería Electrónica

De manera simple ...

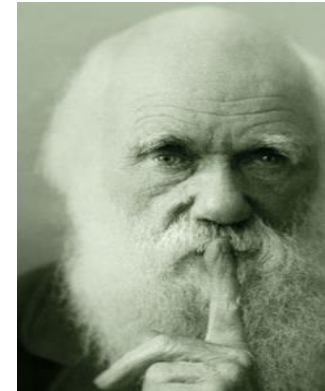
Los individuos que **mejor se adaptan al medio** son aquellos que tienen **más probabilidades de dejar descendencia**, y que por lo tanto, sus genes pasarán a las siguientes generaciones.

La teoría de Darwin también describe que aquellas **modificaciones genéticas** que hacen que los **individuos se adapten mejor al medio**, tienen mayor probabilidad de perdurar en el tiempo.

¿Cómo lo relacionamos con problemas de optimización?

Teoría de la evolución de Darwin

Darwin y optimización



ACE-TI

Grupo de investigación
Ingeniería Electrónica

¿Individuo? → Posible solución al problema

Individuo

X_i	Y_i
-------	-------

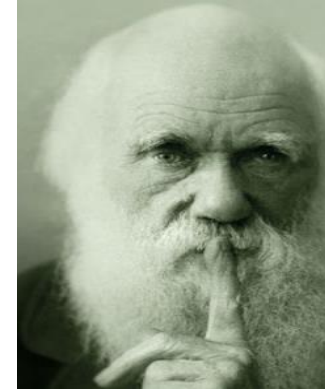
Población

X_1	Y_1
X_2	Y_2
X_3	Y_3
...	...
X_{N-1}	Y_{N-1}
X_N	Y_N

Codificación de las variables del problema como una cadena genes o cromosoma

Teoría de la evolución de Darwin

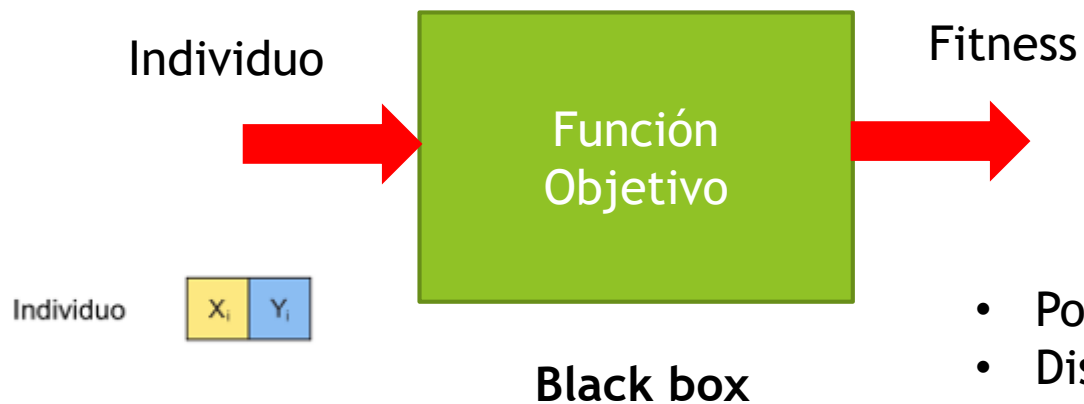
Darwin y optimización



ACE-TI

Grupo de investigación
Ingeniería Electrónica

¿Adaptación al medio? → Desempeño o fitness en la función objetivo

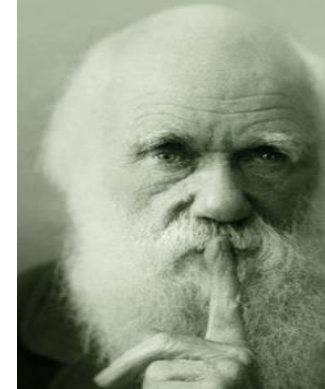


Si el desempeño es alto, se adapta bien al medio (maximizar).

- Potencia
- Distancia recorrida
- Coste

Teoría de la evolución de Darwin

Darwin y optimización

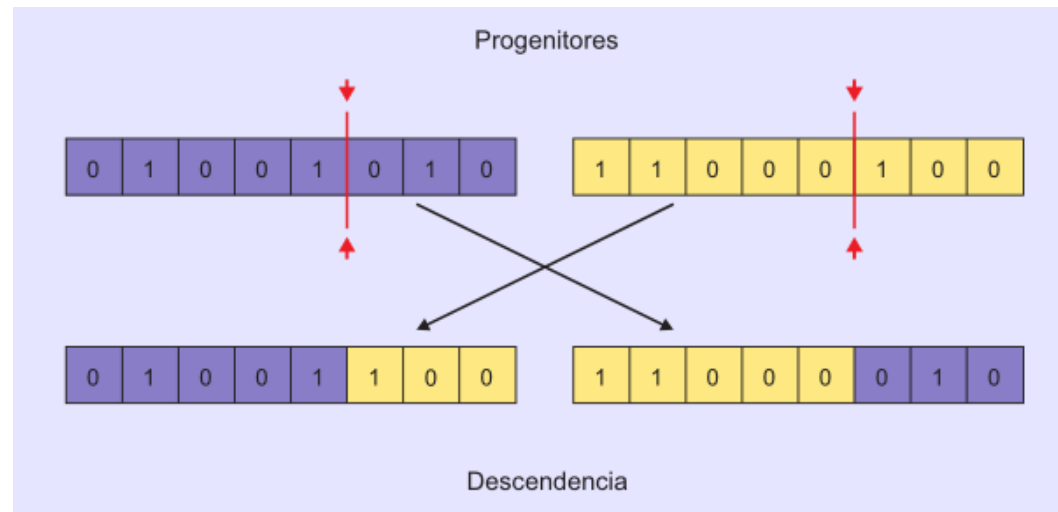


ACE-TI

Grupo de investigación
Ingeniería Electrónica

¿**Modificaciones genéticas?** → Creamos nuevos individuos (descendientes), dos tipos: **cruce** y **mutación**

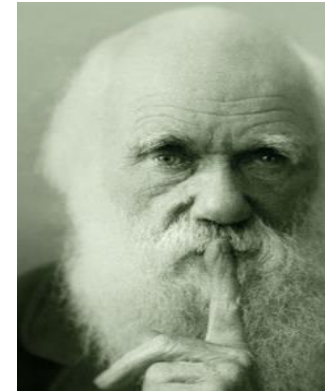
Cruce: de padres se crean dos hijos



Cruce de un punto

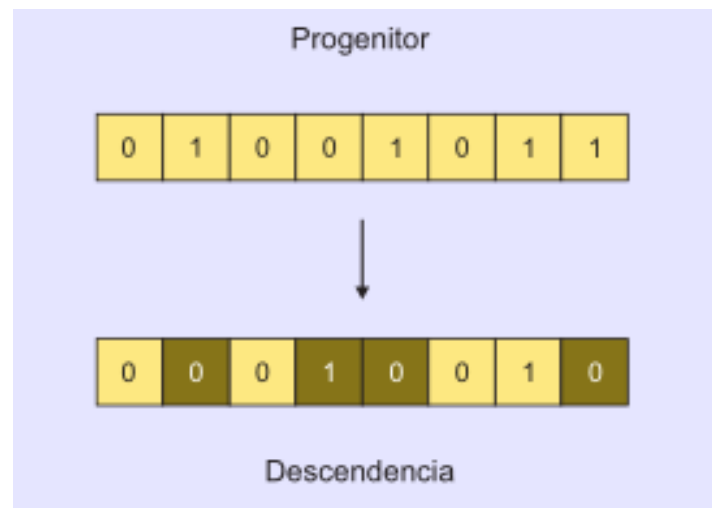
Teoría de la evolución de Darwin

Darwin y optimización



¿Modificaciones genéticas? → Creamos nuevos individuos (descendientes), dos tipos: cruce y mutación

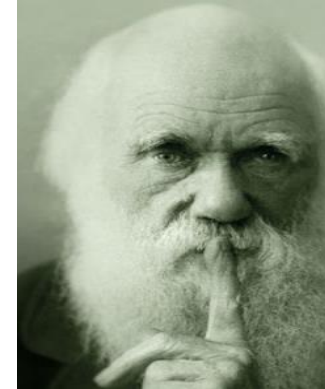
Mutación: mutación de un individuo



BitFlip

Teoría de la evolución de Darwin

Darwin y optimización

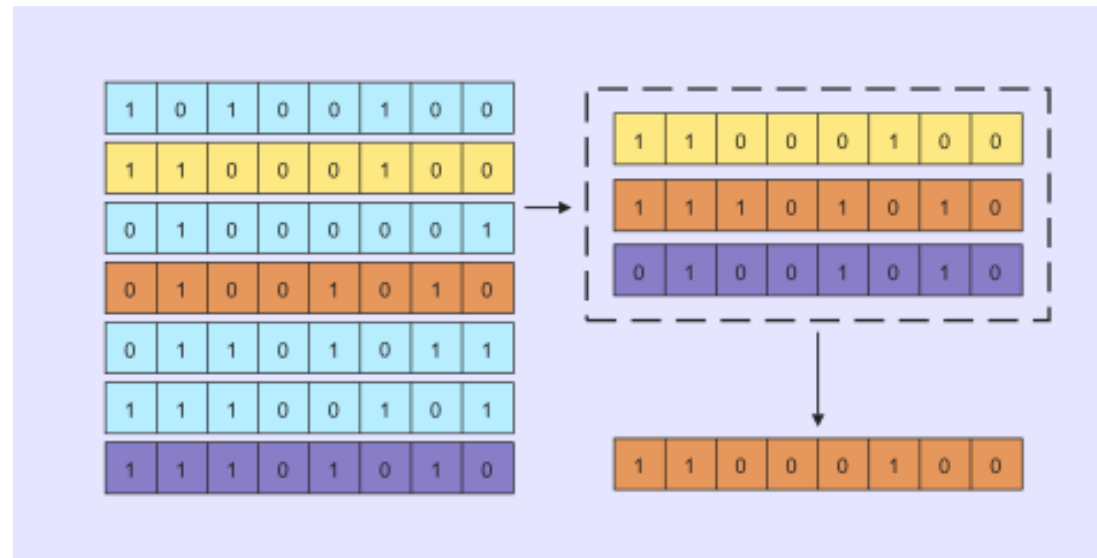


ACE-TI

Grupo de investigación
Ingeniería Electrónica

¿Qué genes perduran en el tiempo? → Selección elitista para elegir los individuos que participarán en la operaciones genéticas

Selección mediante
torneo



**Los mejores individuos
tienen mayor probabilidad
de ser seleccionados.**

Algoritmos genéticos

Idea global de los GAs

- Población de individuos o posible soluciones que evolucionan a lo largo de un número de generaciones, creándose mejores individuos mediante operaciones genéticas: selección, cruce y mutación).



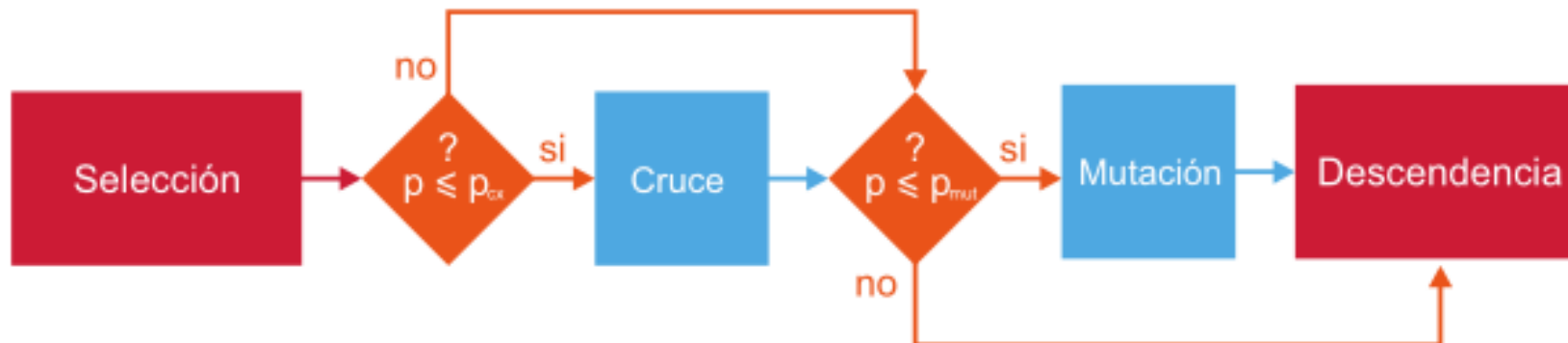
Población
inicial es
aleatoria



Algoritmos genéticos

Características principales

- ▶ GAs utilizan métodos heurísticos, basados en probabilidad (no son métodos exactos) → *Pero sí podemos obtener buenas soluciones!! En muchos casos ni sabes el óptimo.*
 - ▶ Los operadores genéticos son probabilísticos



Contenido día II

- ▶ Resumen del día I.
- ▶ Problemas con múltiples objetivos.
- ▶ Dominancia de Pareto.
- ▶ Algoritmos NSGA-II.
- ▶ El problema de la suma de subconjuntos.
- ▶ Ejemplos de problemas de Ingeniería.
- ▶ Por donde seguir



ACE-TI

Grupo de investigación
Ingeniería Electrónica



Problemas con múltiples objetivos

Optimización planta industrial

- ▶ Ahora tenemos dos objetivos: Maximizar la potencia y disminuir el costo

Parámetros de la planta:

- Válvulas.
- Cintas.
- Motores.



Problemas:

- Maximizar la Potencia
 - Minimizar el coste
- (Objetivos opuestos)



Problemas con múltiples objetivos

Optimización planta industrial

- Solución I: Transformación del problema a un solo objetivo

Parámetros de la planta:

- Válvulas.
- Cintas.
- Motores.



Función objetivo:

$$F = w_1 \times P + w_2 \times C$$

W1 y W2 son pesos que tenemos que ajustar según la importancia que le demos a cada objetivo



Problemas con múltiples objetivos

Optimización planta industrial

- Transformación del problema a un solo objetivo



Función objetivo:

$$F = w_1 \times P + w_2 \times C$$

Si $w_1 = w_2 = 0,5$

Dos posibles soluciones:

$$P_1 = 0,8 \text{ y } C_1 = 0,2$$

$$F_1 = 0,5 \times 0,8 + 0,5 \times 0,2 = 0,5$$

$$P_2 = 0,2 \text{ y } C_2 = 0,8$$

$$F_2 = 0,5 \times 0,2 + 0,5 \times 0,8 = 0,5$$

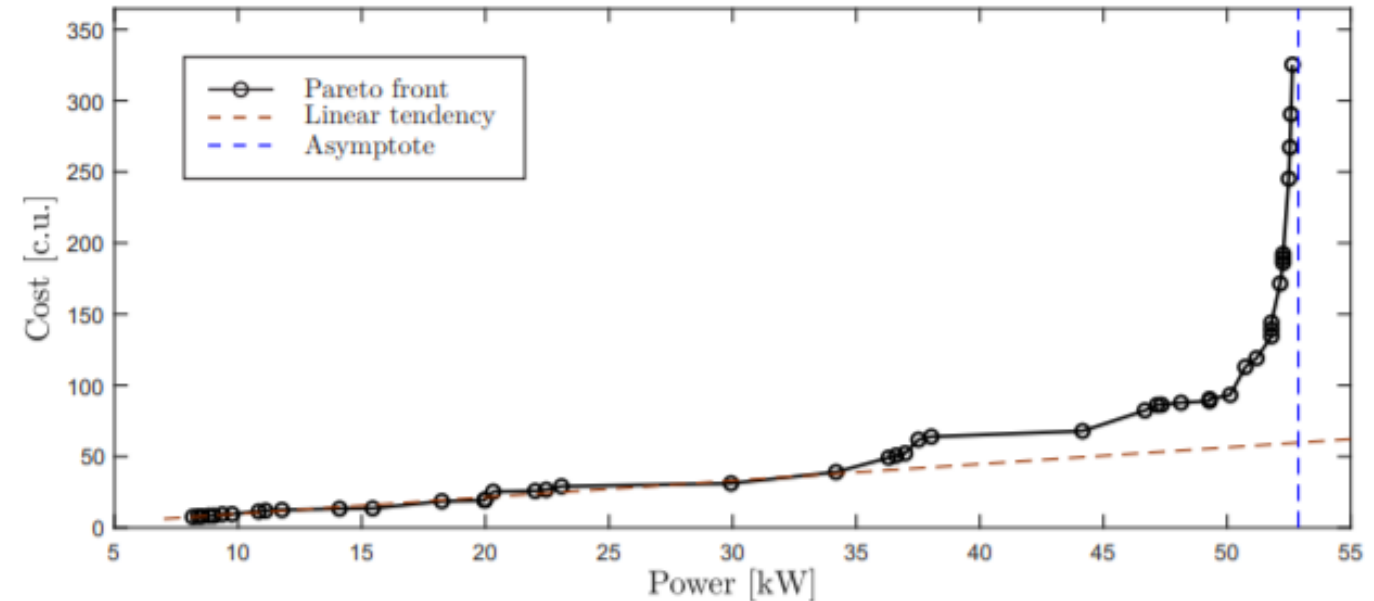
Problema: dos soluciones que son muy distintas tienen el mismo Fitness.



Problemas con múltiples objetivos

Optimización planta industrial

► Solución II: Frente de Pareto



Conjuntos de soluciones que no están dominadas

Contenido día II

- ▶ Resumen del día I.
- ▶ Problemas con múltiples objetivos.
- ▶ Dominancia de Pareto.
- ▶ Algoritmos NSGA-II.
- ▶ El problema de la suma de subconjuntos.
- ▶ Ejemplos de problemas de Ingeniería.
- ▶ Por donde seguir.



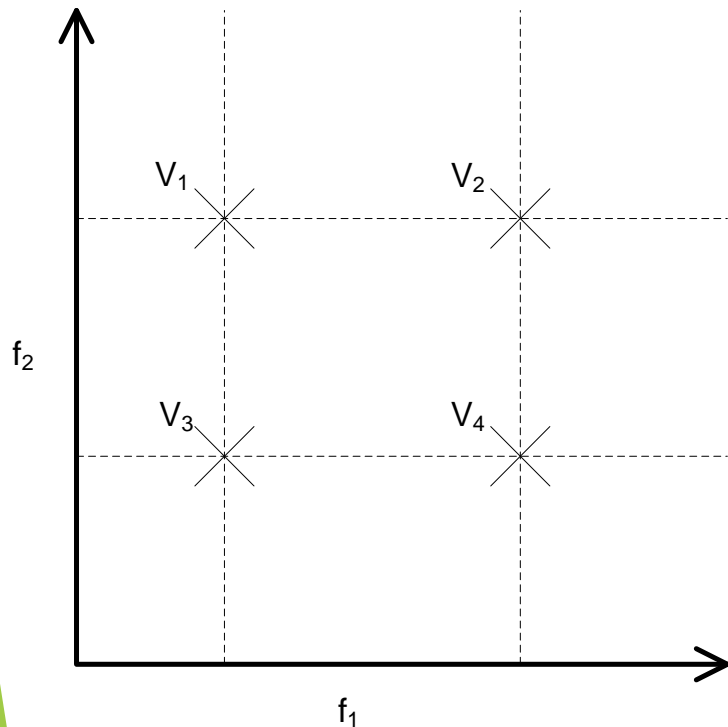
ACE-TI

Grupo de investigación
Ingeniería Electrónica



Problemas con múltiples objetivos

Doninancia de Pareto

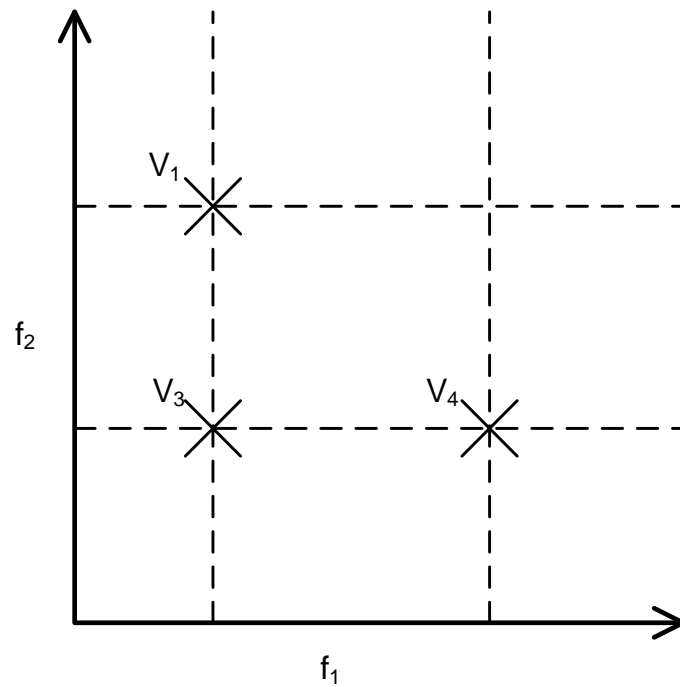


- **Objetivos:** minimizar f_1 y f_2 .
- **Dominancia de Pareto:** Una solución domina a otra si es **estrictamente mejor** que la otra en todos los objetivos.
- Cuatro posibles soluciones: V_1 , V_2 , V_3 y V_4 .
 - $V_1 < V_2$ en f_1 , $V_1 = V_2$ en f_2 (V_1 no domina V_2)
 - $V_1 = V_3$ en f_1 , $V_3 < V_1$ en f_2 (V_1 no domina V_3)
 -
 - $V_3 < V_2$ en f_1 , $V_3 < V_2$ en f_2 (V_3 domina V_2)

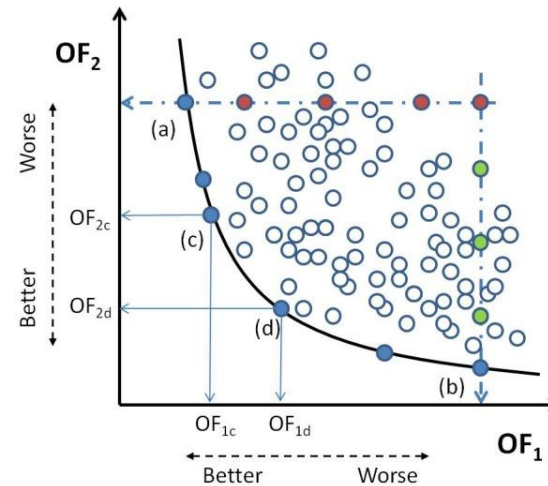


Problemas con múltiples objetivos

Frente de Pareto



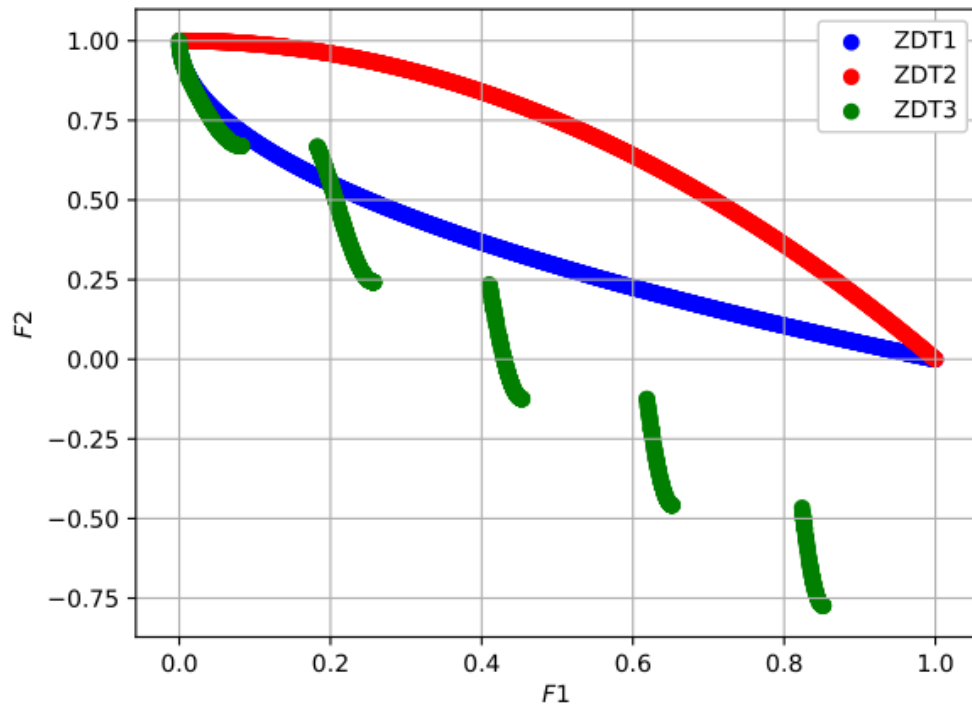
Soluciones no dominadas





Problemas con múltiples objetivos

Tipos de frente de Pareto

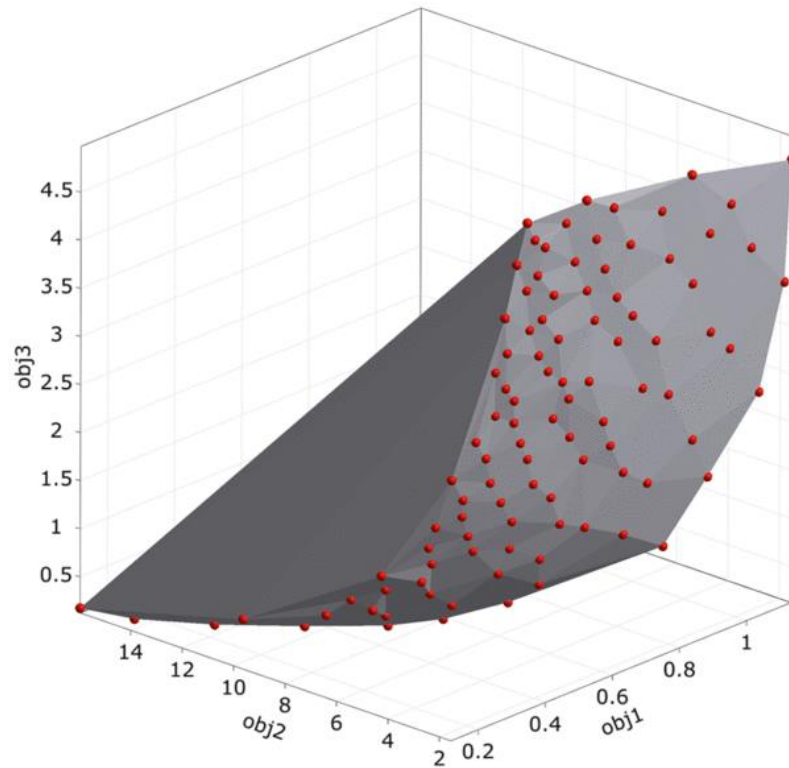


Cóncavo, convexo y discontinuo.



Problemas con múltiples objetivos

Tipos de frente de Pareto



Problemas con tres objetivos.

Con más de tres objetivos ya no puedo Representarlo. Se pueden representar sus proyecciones dos a dos.



Problemas con múltiples objetivos

Objetivo

- ▶ Algoritmo genético que nos proporcione el frente de Pareto de nuestro problema de manera eficiente (rápido).
- ▶ Problema: Tenemos que realizar muchas comparaciones. Con M objetivo y N individuos \rightarrow Complejidad $O(M \cdot N^3)$.
- ▶ Solución: NSGA-II.

Contenido día II

- ▶ Resumen del día I.
- ▶ Problemas con múltiples objetivos.
- ▶ Dominancia de Pareto.
- ▶ Algoritmos NSGA-II.
- ▶ El problema de la suma de subconjuntos.
- ▶ Ejemplos de problemas de Ingeniería.
- ▶ Por donde seguir.



ACE-TI

Grupo de investigación
Ingeniería Electrónica

Algoritmos con múltiples objetivos

Nondominated Sorting Genetic Algorithm II (NSGA-II):

Es uno de los algoritmos genéticos más utilizados en investigación. Proporciona buenos resultados y es fácil de implementar.

Propuesto en:

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A Fast and Elistist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation, Vol. 6, 2002.

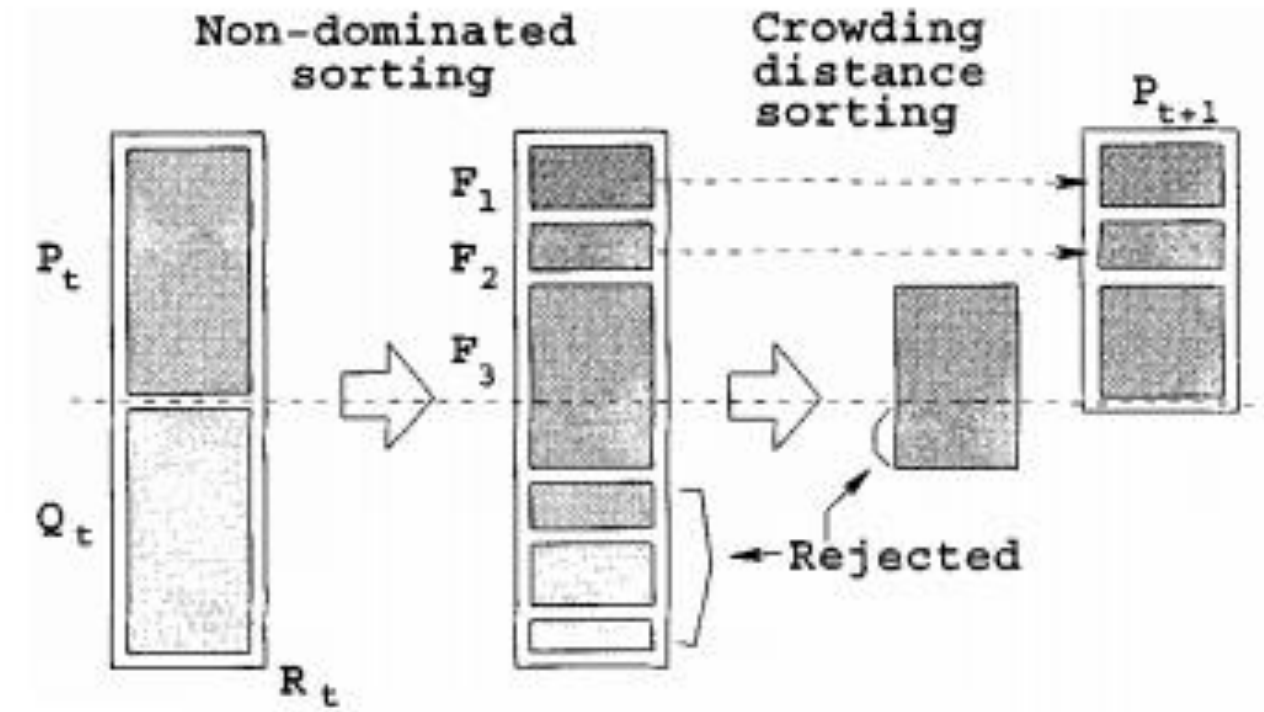
- Fast: Porque se propone un algoritmo para comparar las soluciones del Pareto front de forma rápida.
- Garantizar que dos soluciones en el frente de Pareto estén distanciadas.

NSGA-II

Selección de individuos: 1) Consideramos la población actual y la previa, 2) Ordenamos por ranking basado en dominancia y 3) aplicamos la condiciones de distancia.

Ranking de dominancia

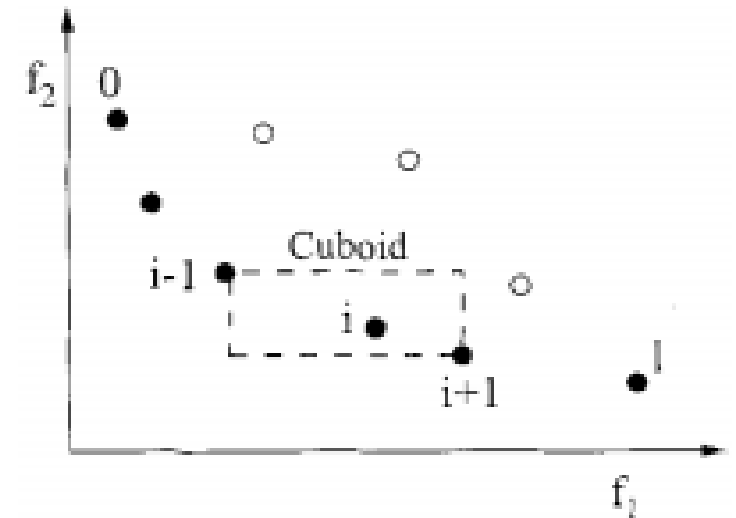
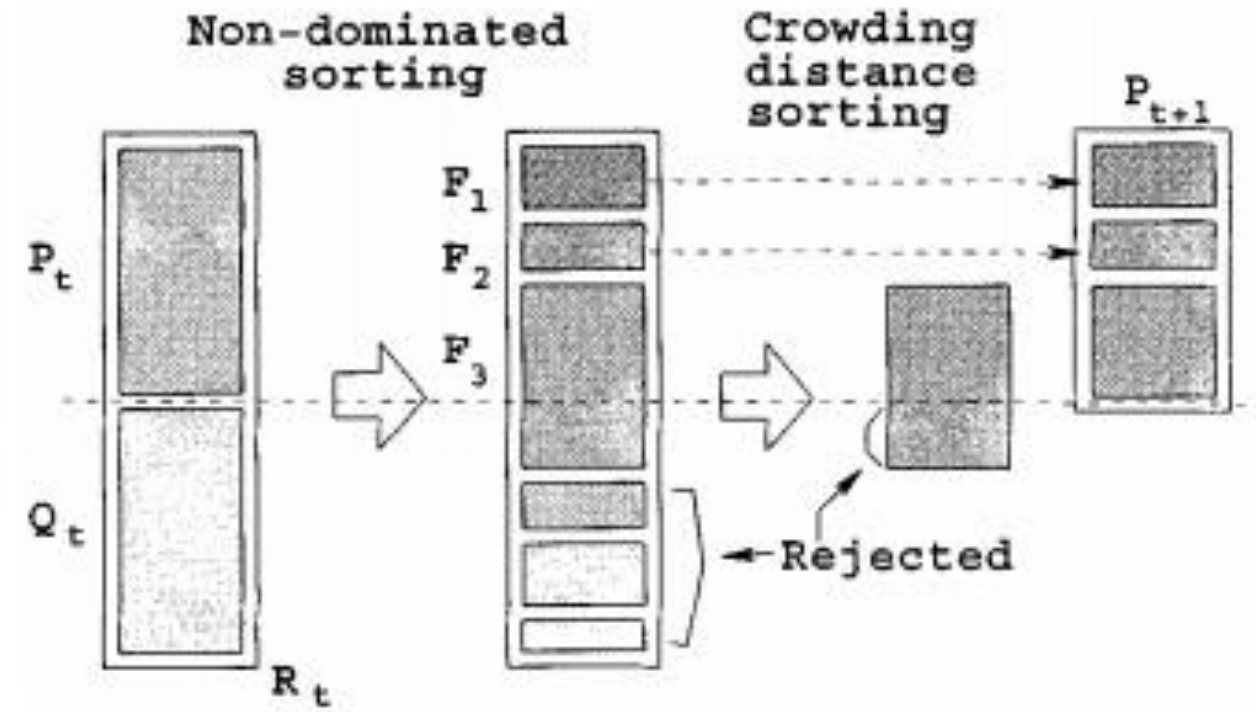
- F1: soluciones dominantes.
- F2: soluciones que solo están dominadas por alguna solución de F1.
- ...



NSGA-II

Selección de individuos: 1) Consideramos la población actual y la previa, 2) Ordenamos por ranking basado en dominancia y 3) aplicamos la condiciones de distancia.

Métrica de distancia



Contenido día II

- ▶ Resumen del día I.
- ▶ Problemas con múltiples objetivos.
- ▶ Dominancia de Pareto.
- ▶ Algoritmos NSGA-II.
- ▶ El problema de la suma de subconjuntos.
- ▶ Ejemplos de problemas de Ingeniería.
- ▶ Por donde seguir.



ACE-TI

Grupo de investigación
Ingeniería Electrónica



Suma de subconjuntos

Problema:

- ▶ Dado un conjunto de elementos finito, queremos obtener el subconjunto mínimo que sume cierta cantidad. No nos podemos pasar (restricción).
 - ▶ Objetivo 1: Minimizar la diferencia de la suma de los elementos del subconjunto.
 - ▶ Objetivo 2: Minimizar el número de elementos del subconjunto.
- ▶ Problema NP-duro. Complejidad exponencial.
- ▶ Tenemos que definir:
 - ▶ Conjunto de valores.

20	3	0	23	8	7	24	4	28	17	2	18	13	1	27	16	21	29	25	9	15	11	12	5
6	22	26	10	14	19																		
 - ▶ Suma objetivo → Tiene que ser factible, en nuestro caso todos los elementos suman 435 → Vamos a considerar una suma objetivo de 333.



Suma de subconjuntos

Representación de los individuos: (posibles soluciones)

- **Lista de elementos que se seleccionarán del conjunto original.**
- Cada posición puede tener dos valores: 0 no se incluye el elemento, 1 se incluye el elemento.
- Necesitamos una función para generar individuos aleatorios (población)



Ejemplo individuo





Suma de subconjuntos

Población: Conjunto de individuos sobre los que se aplican los operadores genéticos.

- Población inicial:
 - Aleatoria o dada.
 - Tamaño y diversidad.

X_1	X_2	X_3	...	X_{n-1}	X_n
X_1	X_2	X_3	...	X_{n-1}	X_n
X_1	X_2	X_3	...	X_{n-1}	X_n
X_1	X_2	X_3	...	X_{n-1}	X_n



Problema del viajero (TSP)

Función de fitness: medida de la bondad de cada individuo de una población. **Distancia que recorre el agente.**

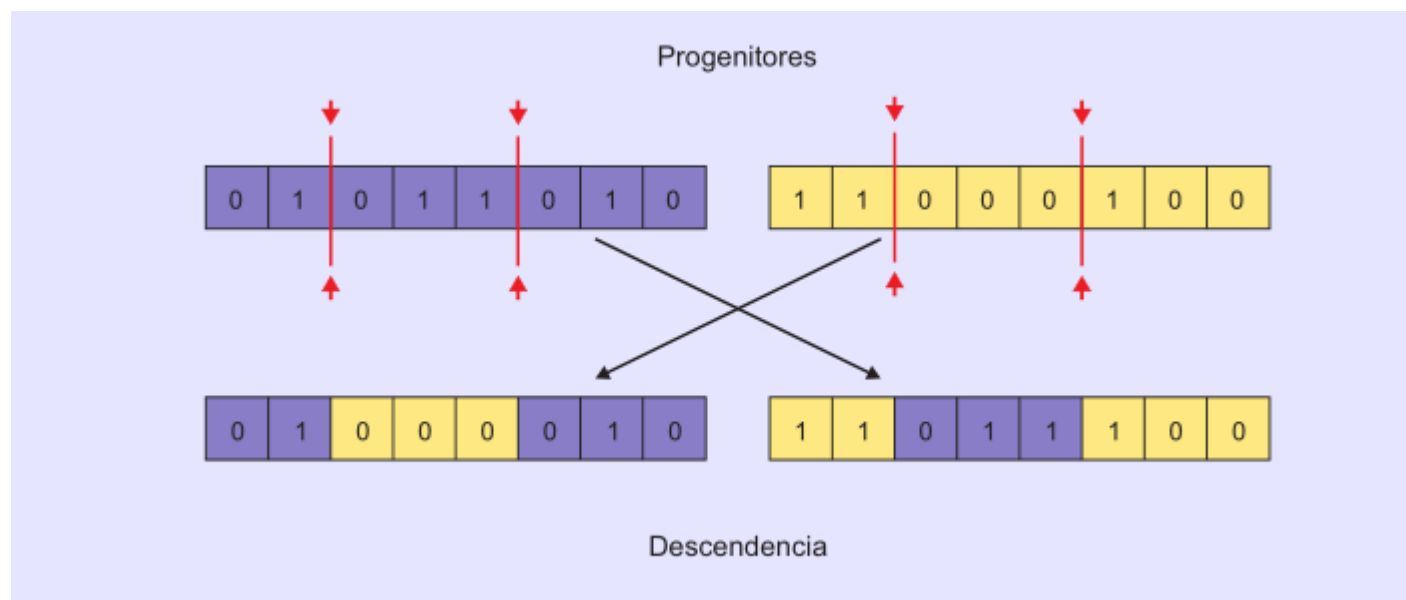
```
37 def funcion_objetivo(individuo, suma_objetivo):
38     """
39     Función objetivo para el problema de la suma de subconjuntos
40     Entrada: Recibe como entrada el individuo y la suma objetivo
41     que se tiene que satisfacer con la suma de los elementos del subconjunto.
42     Salida: Como objetivo 1 se devuelve el número de elementos del
43     subconjunto. Como objetivo 2 se devuelve la diferencia con respecto
44     a la suma objetivo.
45     """
46     subconjunto = CONJUNTO[np.array(individuo) == 1]
47     suma_subconjunto = np.sum(subconjunto)
48     diferencia = suma_objetivo - suma_subconjunto
49     n_elementos = sum(individuo)
50     if diferencia < 0: # nos pasamos
51         return 10000, 10000 # pena de muerte
52     if n_elementos == 0: # no se selecciona ninguna elemento
53         return 10000, 10000
54     return n_elementos, diferencia
```



Problema del viajero (TSP)

Operadores genéticos: construcción de la siguiente generación

- Cruce: dos individuos padres combinan su información para generar dos individuos hijos.
- **Cruce de dos puntos:**

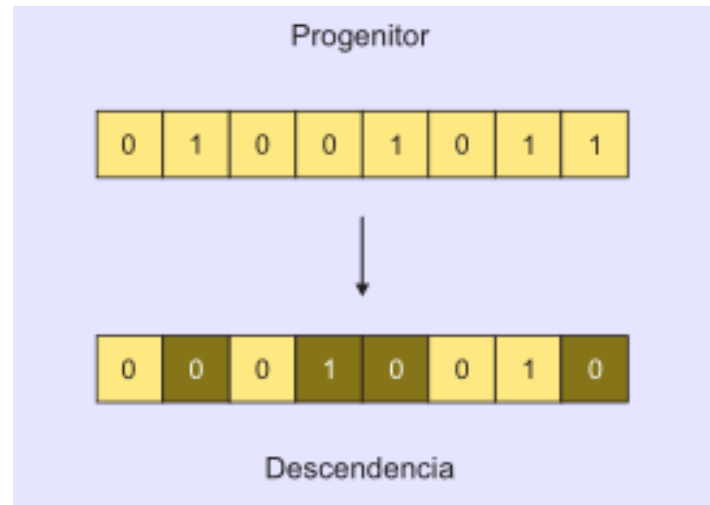




Problema del viajero (TSP)

Operadores genéticos: construcción de la siguiente generación.

- ▶ **Mutación:** alteración de la información genética de un individuo al pasar a la siguiente generación.
- ▶ **Bitflip:**

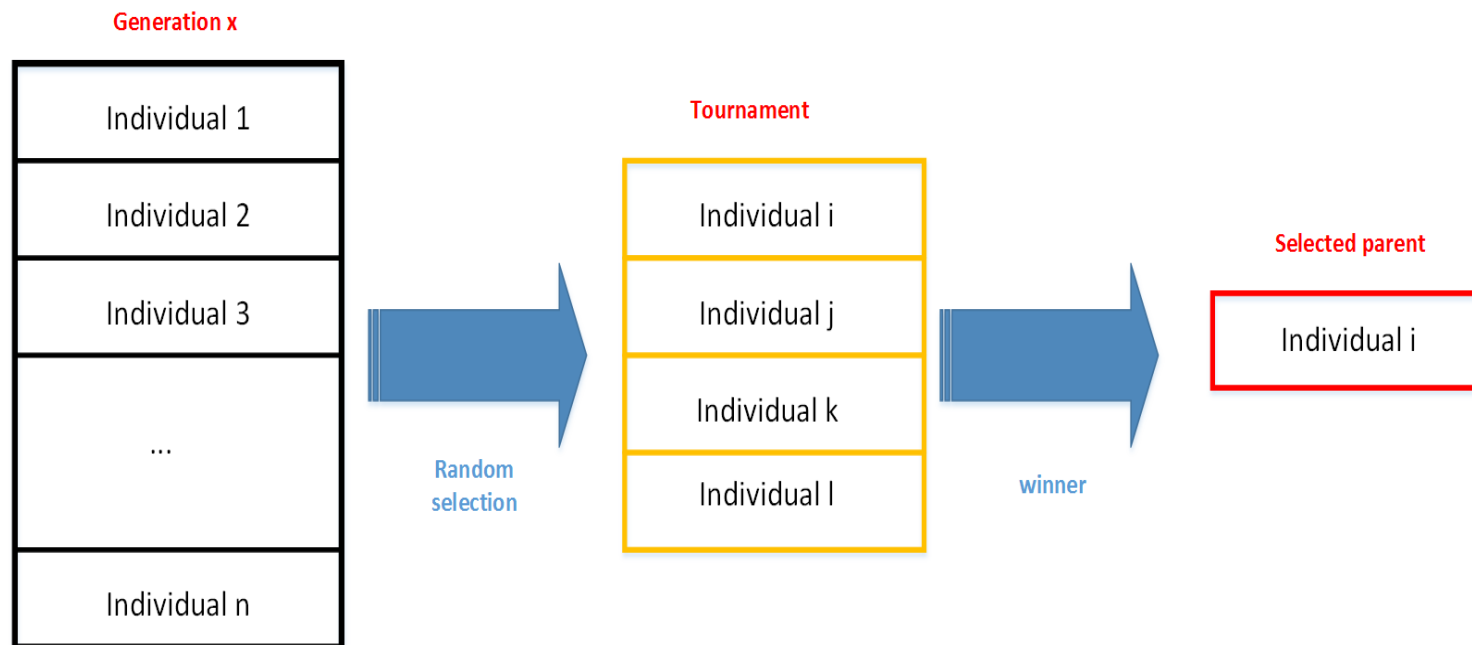




Problema del viajero (TSP)

Selección: ¿Cómo seleccionamos a los padres?

- Selección mediante torneo → Torneo de tamaño 3 suele funcionar bien.



Contenido día II

- ▶ Resumen del día I.
- ▶ Problemas con múltiples objetivos.
- ▶ Dominancia de Pareto.
- ▶ Algoritmos NSGA-II.
- ▶ El problema de la suma de subconjuntos.
- ▶ Ejemplos de problemas de Ingeniería.
- ▶ Por donde seguir.



ACE-TI

Grupo de investigación
Ingeniería Electrónica

Aplicaciones

Aplicaciones: planificación de rutas de vehículos autónomos

Lago Ypacarai (Asunción, Paraguay)

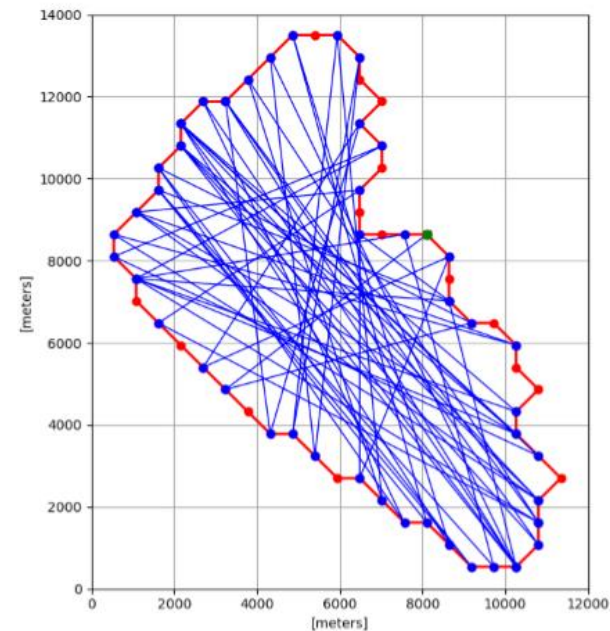


Catamarán autónomo



Múltiples Objetivos:

-



Aplicaciones

Despacho económico en redes eléctricas

Microred



Elementos.

1. Generador eólico.
2. Paneles fotovoltaicos
3. Microturbina
4. Generador Diesel
5. Sistema de almacenamiento de energía (baterías).
6. Viviendas (demanda).



ACE-TI

Grupo de investigación
Ingeniería Electrónica

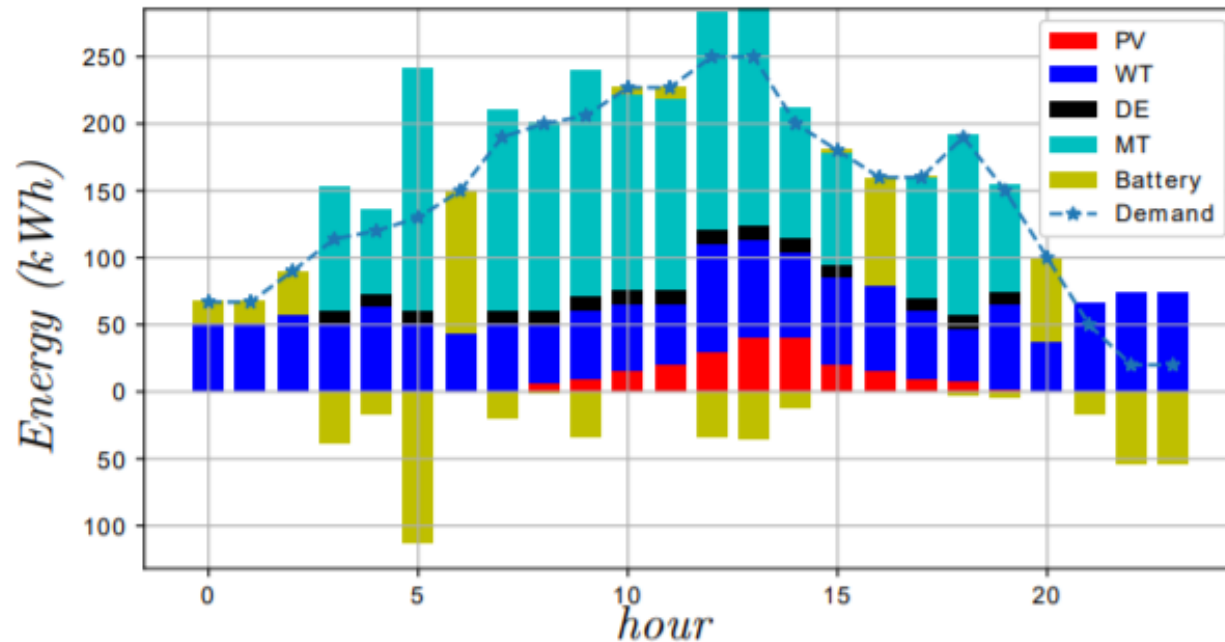
Aplicaciones

Despacho económico redes eléctricas

Múltiples Objetivos:

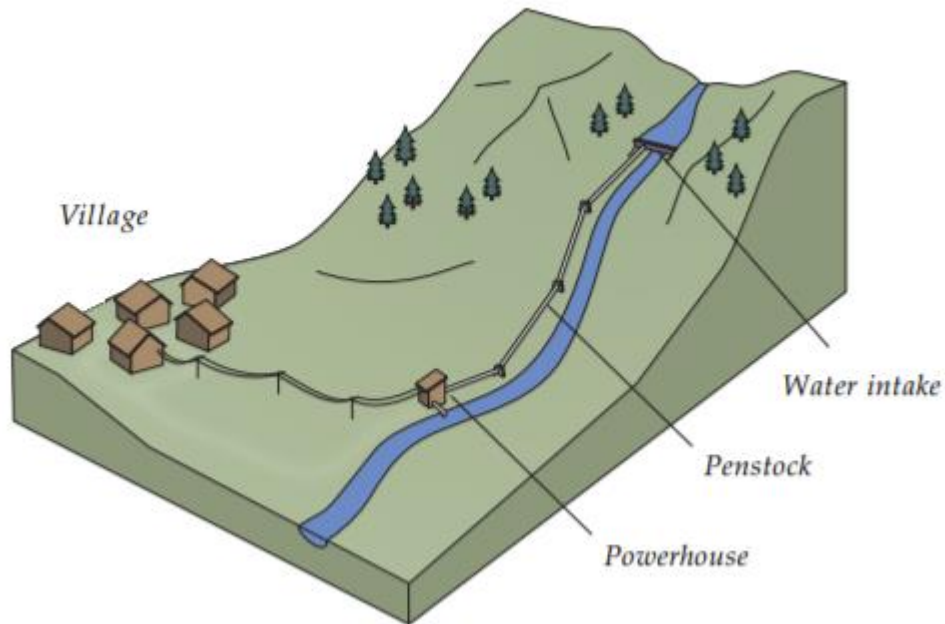
Objetivo 1: Minimizar el coste de operación

Objetivo 2: Maximizar la confiabilidad



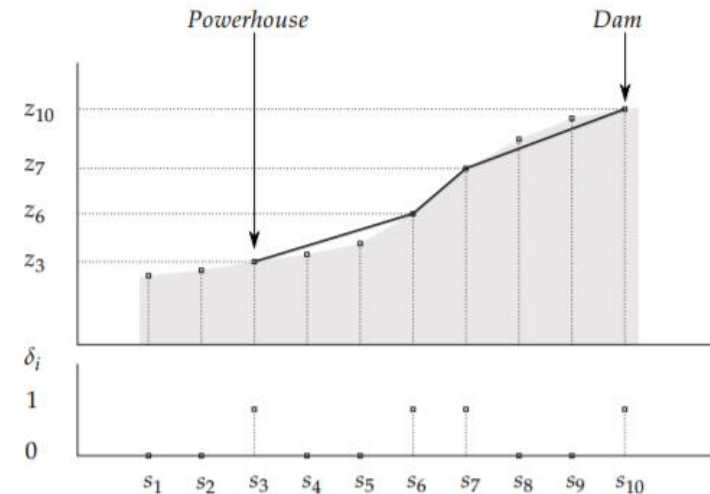
Aplicaciones

Diseño mico planta hidráulica



Dado el perfil del río, debemos decidir:

- Dónde colocar la presa.
- Dónde colocar la tuberías.
- Dónde colocar al casa de máquinas.



Aplicaciones

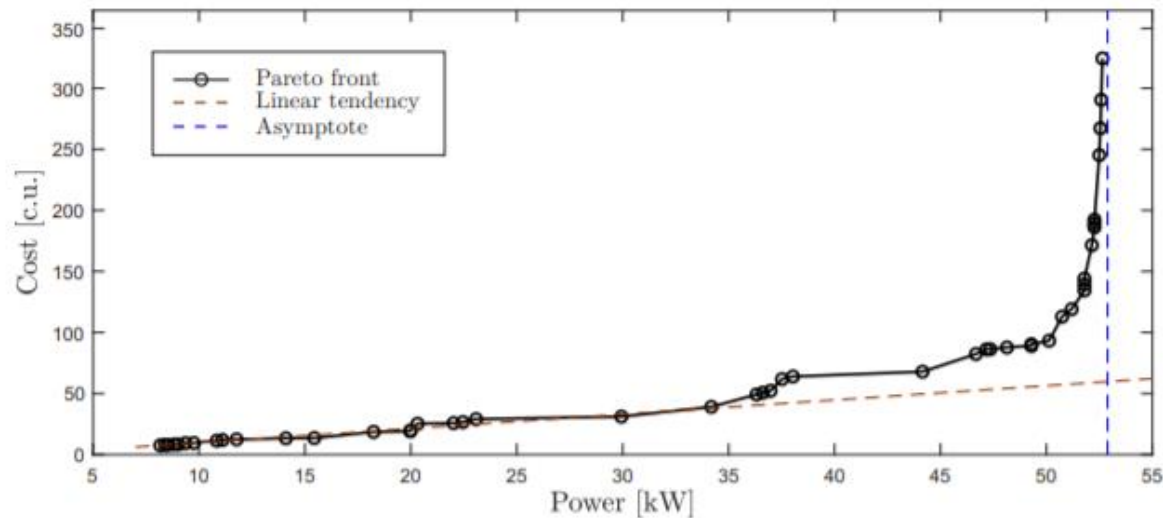
Diseño mico planta hidráulica

Múltiples Objetivos:

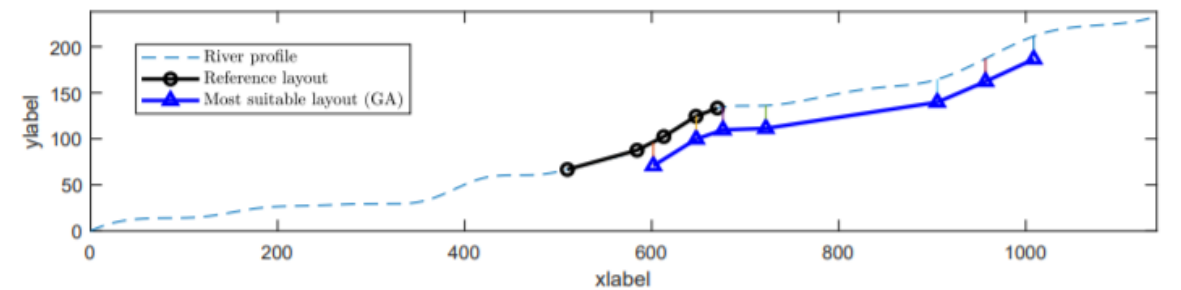
Objetivo 1: Minimizar el coste de operación.

Objetivo 2: Maximizar la potencia eléctrica generada.

Frente de Pareto



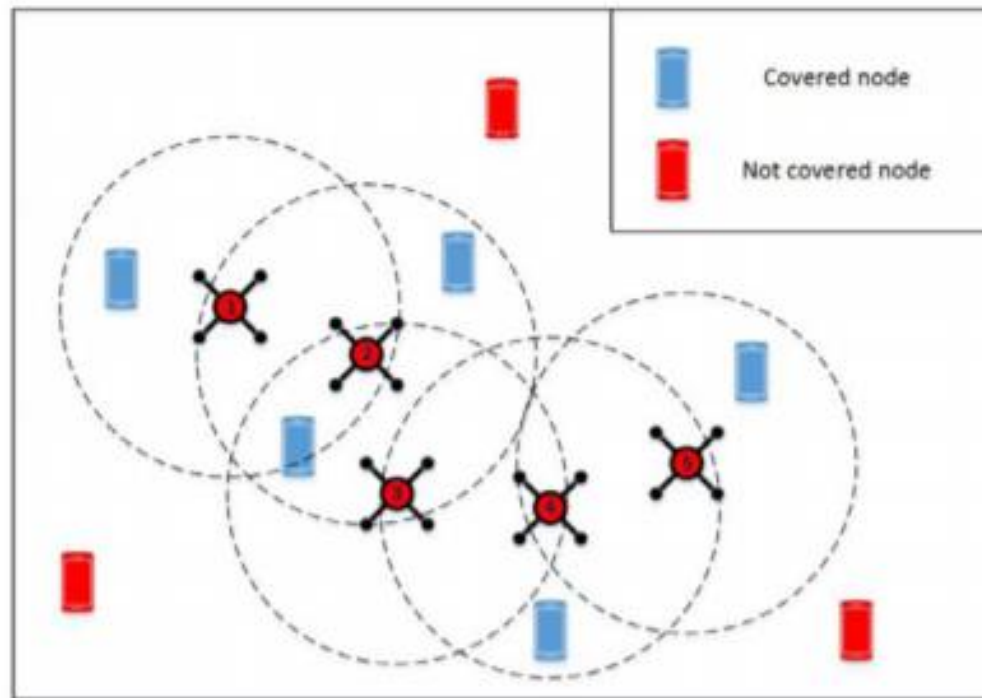
Ejemplo de una solución



ACE-TI

Grupo de investigación
Ingeniería Electrónica

Posicionamiento de sensores (Drones aéreos)



Dado un conjunto de puntos que se desean cubrir, debemos decidir:

- Posiciones de los drones para cubrir los puntos de interés.
- Los drones tienen que formar una red conectada.

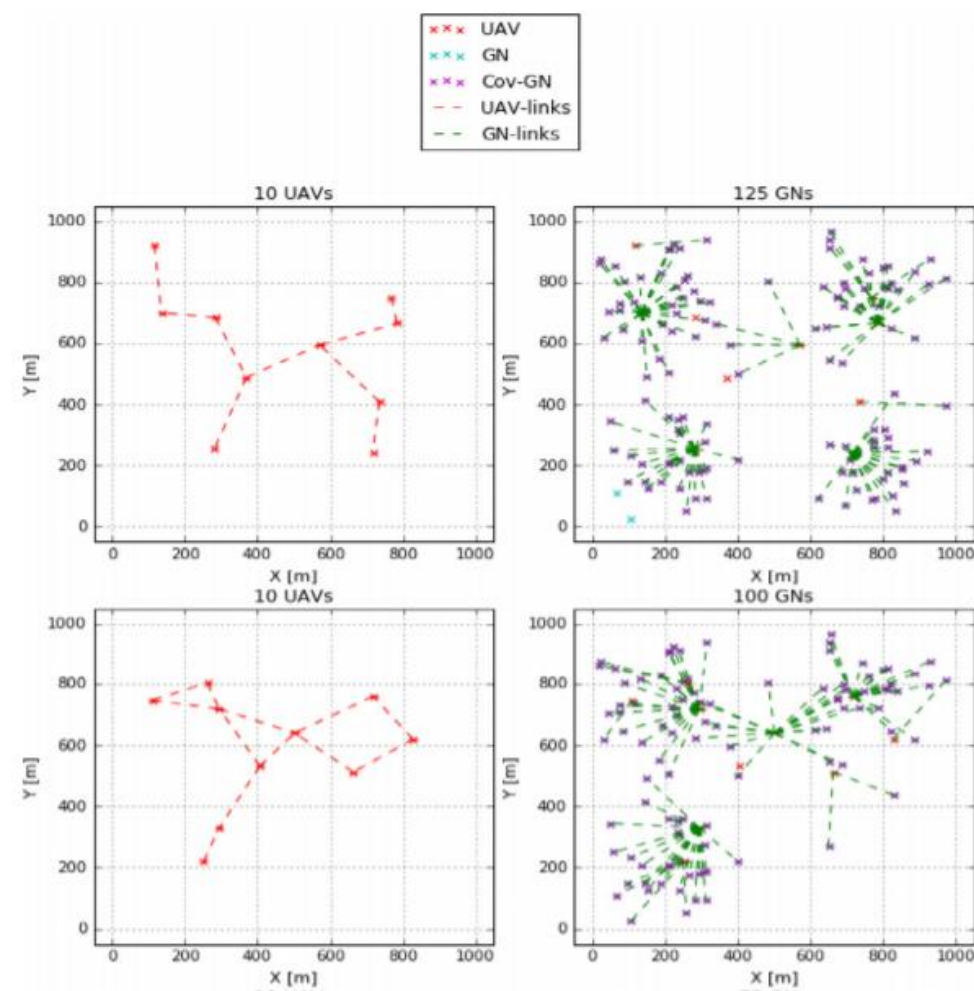
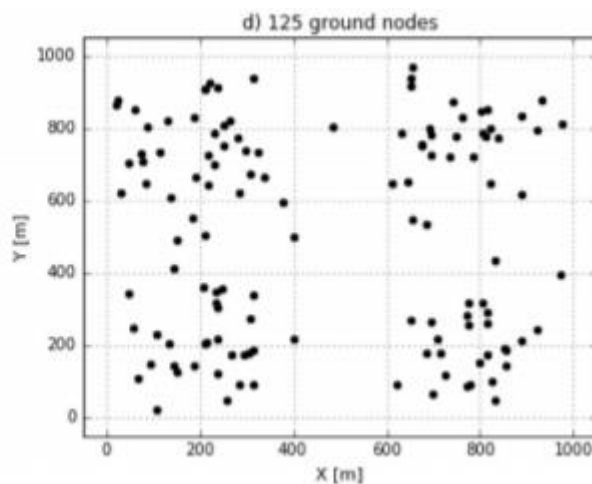
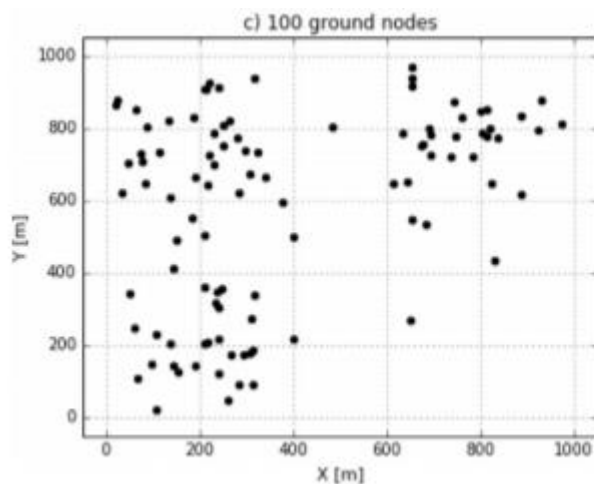
Objetivo: Maximizar el número de puntos cubiertos.

$$F = w_1CO + w_2FTO + w_3RO$$

Posicionamiento de sensores (Drones aéreos)

Soluciones

Escenarios



Contenido día II

- ▶ Resumen del día I.
 - ▶ Problemas con múltiples objetivos.
 - ▶ Dominancia de Pareto.
 - ▶ Algoritmos NSGA-II.
 - ▶ El problema de la suma de subconjuntos.
 - ▶ Ejemplos de problemas de Ingeniería.
- ▶ Por donde seguir.



ACE-TI

Grupo de investigación
Ingeniería Electrónica

Bibliografía

► The Essential of Metaheuristics

Sean Luke

Department of Computer Science
George Mason University

Second Edition

Online Version 2.2
October, 2015



ACE-TI

Grupo de investigación
Ingeniería Electrónica

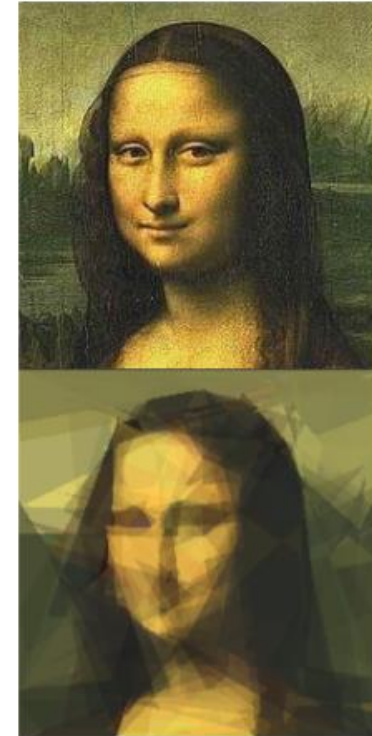


Figure 0 The Mona Lisa, estimated with the $(5 + 1)$ Evolution Strategy. The objective is to find a set of fifty polygons which most closely approximates the original image. After Roger Alsing.

Curso que impartimos en la ESI



ACE-TI

Grupo de investigación
Ingeniería Electrónica

Python: Machine Learning, Optimización y Aplicaciones (IV edición)

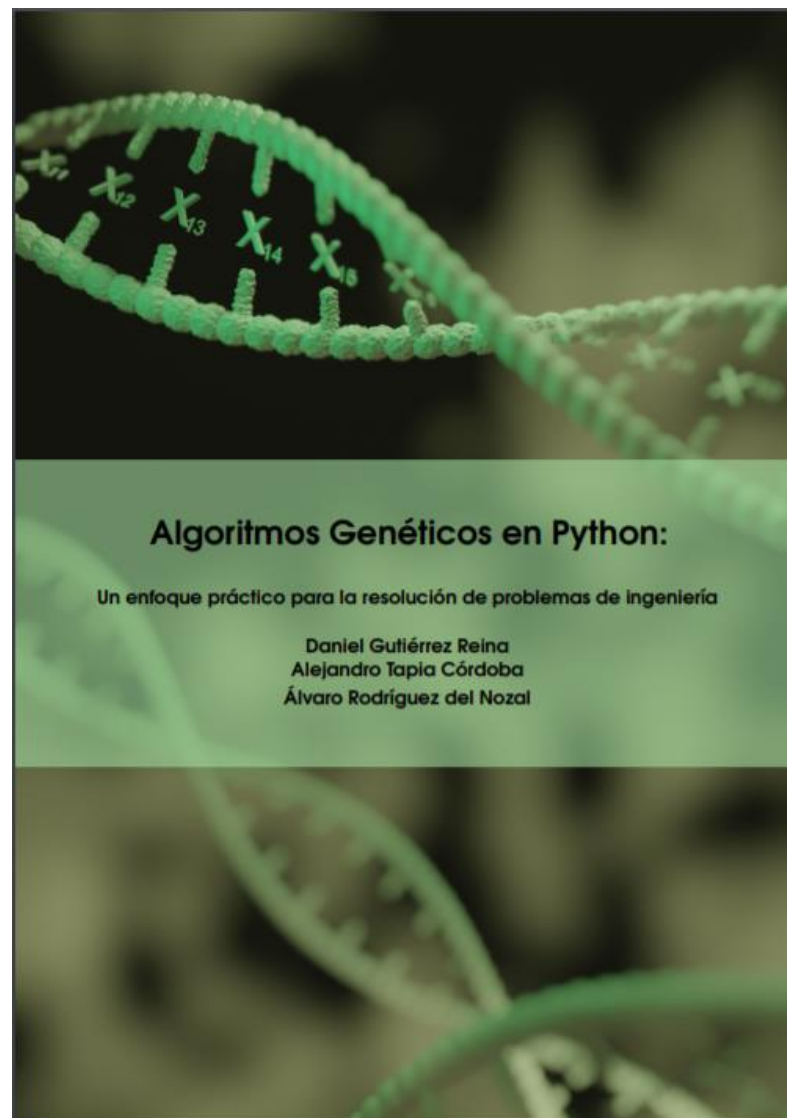
- ▶ Módulo 1: Conocimientos Básicos de Python y sus Módulos Principales. (20 horas).
- ▶ Módulo 2: Machine Learning en Python: Regresión, Clasificadores y Clustering. (16 horas).
- ▶ Módulo 3: Técnicas de Optimización en Python. (16 horas).
- ▶ Módulo 4: Deep Learning con TensorFlow y Keras. (16 horas).
- ▶ Módulo 5: Aplicaciones. (16 horas).



CFP Centro de Formación
Permanente
Vicerrectorado de Ordenación Académica

<https://cfp.us.es/cursos/fc/python-machine-learning-optimizacion-y-aplicaciones/4012/>

Próximamente disponible



ACE-TI

Grupo de investigación
Ingeniería Electrónica

Avisaré por twitter

MUCHAS GRACIAS

Daniel Gutiérrez Reina, dgutierrezreina@us.es