

Algoritmos de optimización bioinspirados

Presentación del Módulo

Inteligencia Artificial
INFO257

Profesor: Jorge Maturana
jorge.maturana@inf.uach.cl

Generalidades

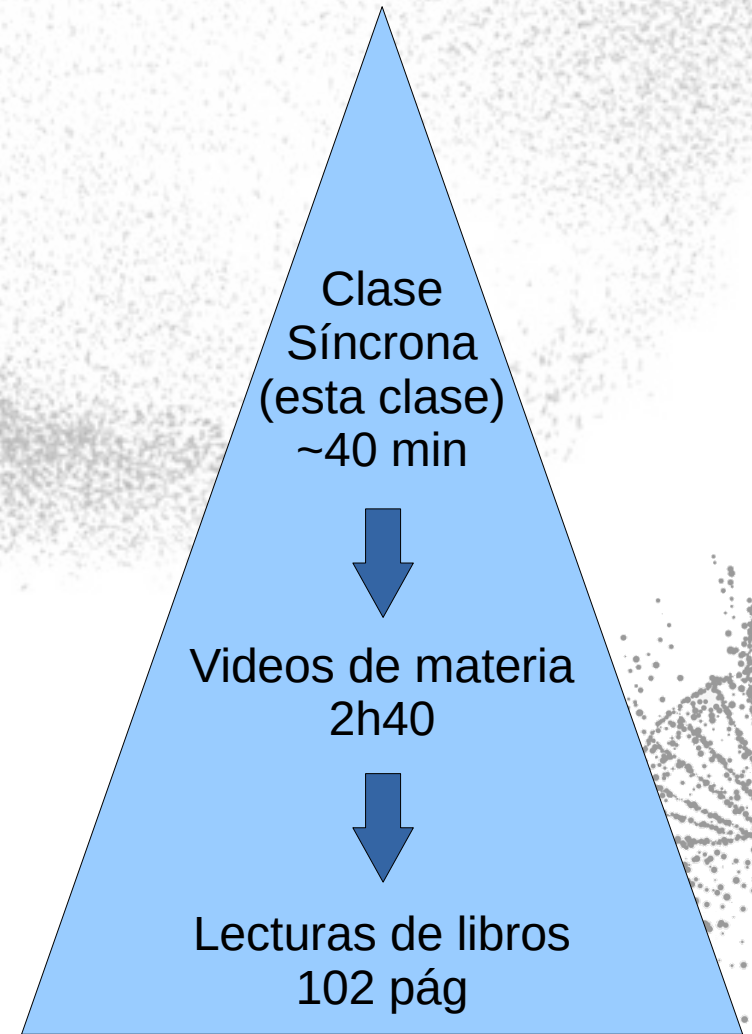
- Propósito:
 - Conocer y aplicar algoritmos de optimización bioinspirados
- Contenidos:
 - Contexto: necesidad de optimización global
 - Metaheurísticas
 - Particle Swarm Optimization
 - Algoritmos Evolutivos
- Evaluación:
 - Trabajo grupal (3-4 personas)
 - Prueba individual

Resultados de aprendizaje

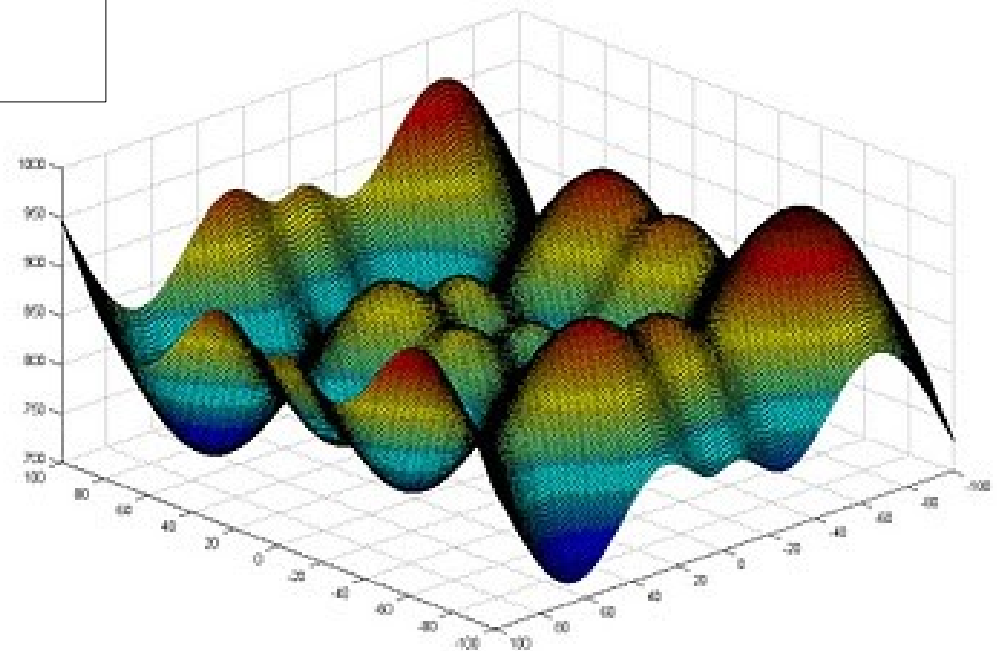
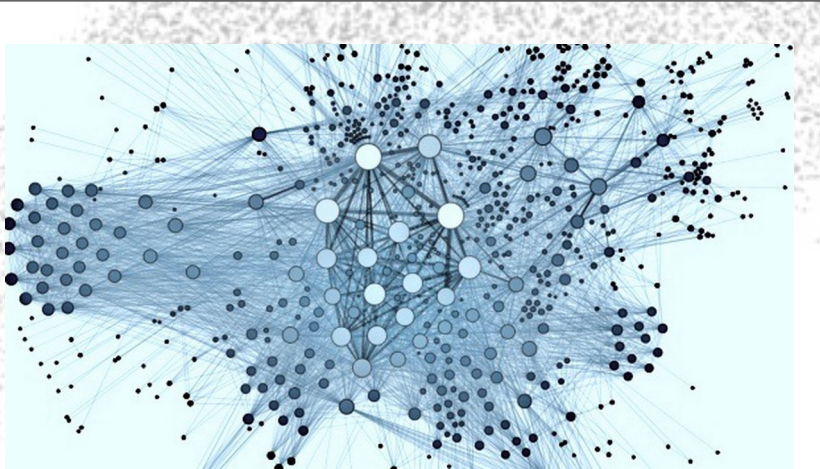
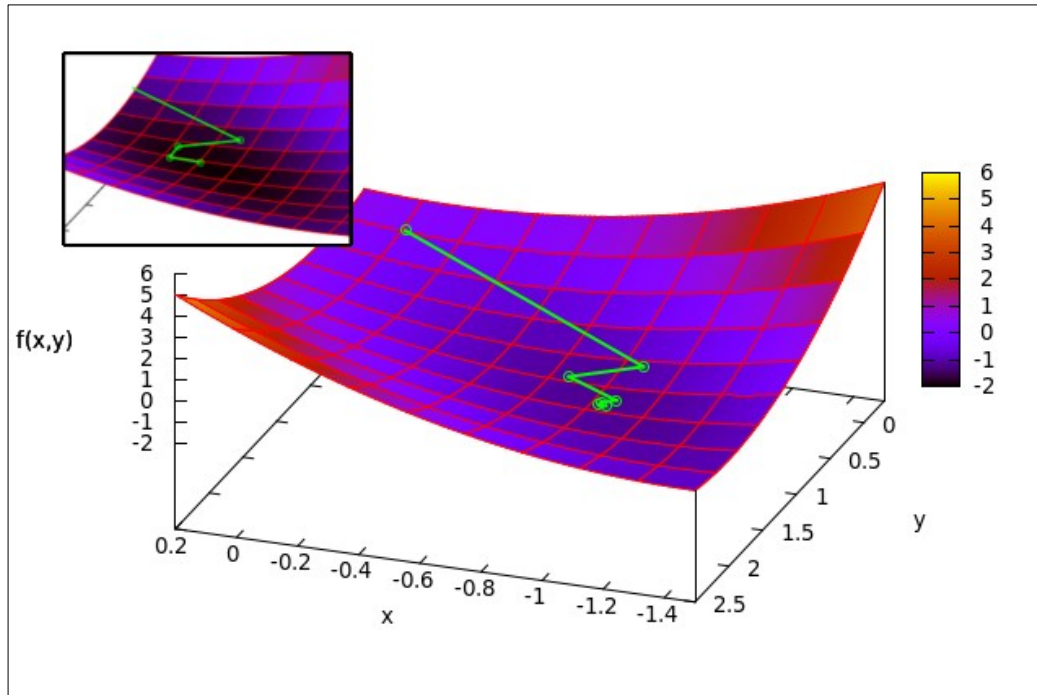
- **Comprender los conceptos asociados a la optimización con metaheurísticas**, tales como función de evaluación, fitness, codificación de soluciones, multimodalidad, espacio de búsqueda, operadores de modificación de soluciones (incluyendo mutación y cruzamiento), vecindad, sesgo de selección, reinserción, explosión combinatoria, exploración, explotación, diversidad poblacional, convergencia y convergencia prematura.
- **Explicar el funcionamiento de los algoritmos PSO y EA.**
- **Reconocer conceptos comunes a distintas metaheurísticas**, tales como exploración, explotación, convergencia y diversidad.
- **Relacionar los conceptos anteriores**, pudiendo predecir el impacto que podría tener la modificación de los parámetros asociados a ellos sobre el funcionamiento del algoritmo.

Estructura de entrega de contenidos

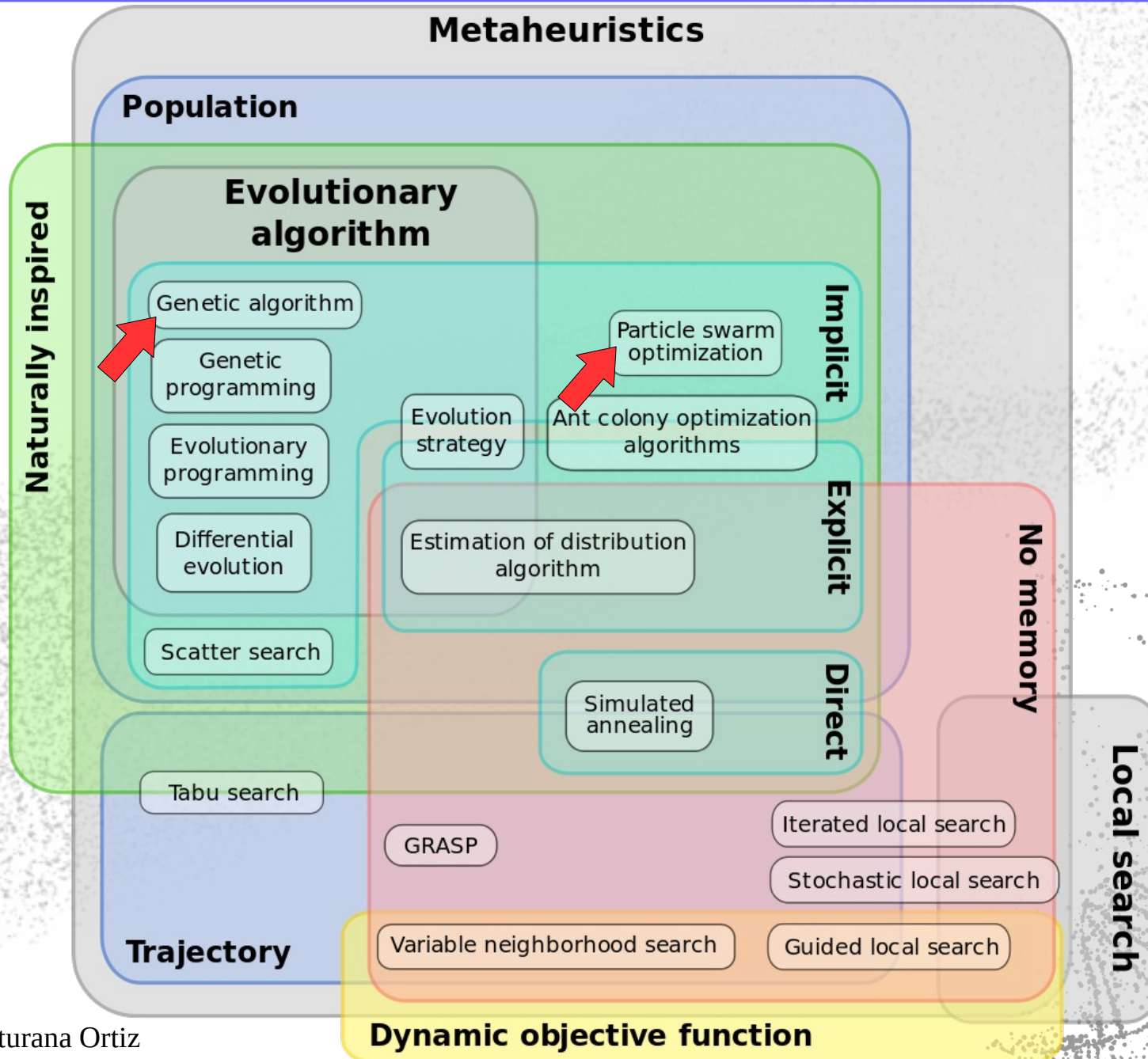
- Contribuye a comprender los conceptos contextualizadamente
- De lo general:
 - Menos información, más general
- A lo particular:
 - Más información, más específica



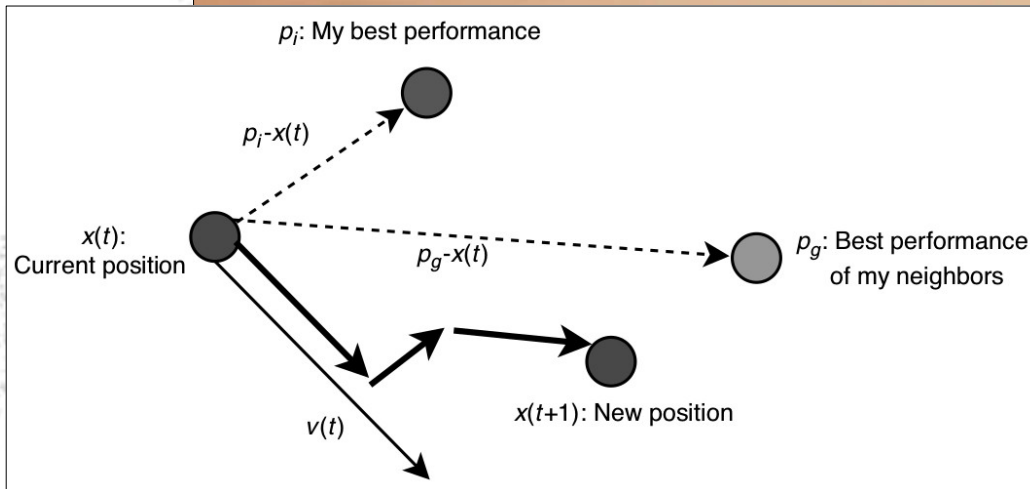
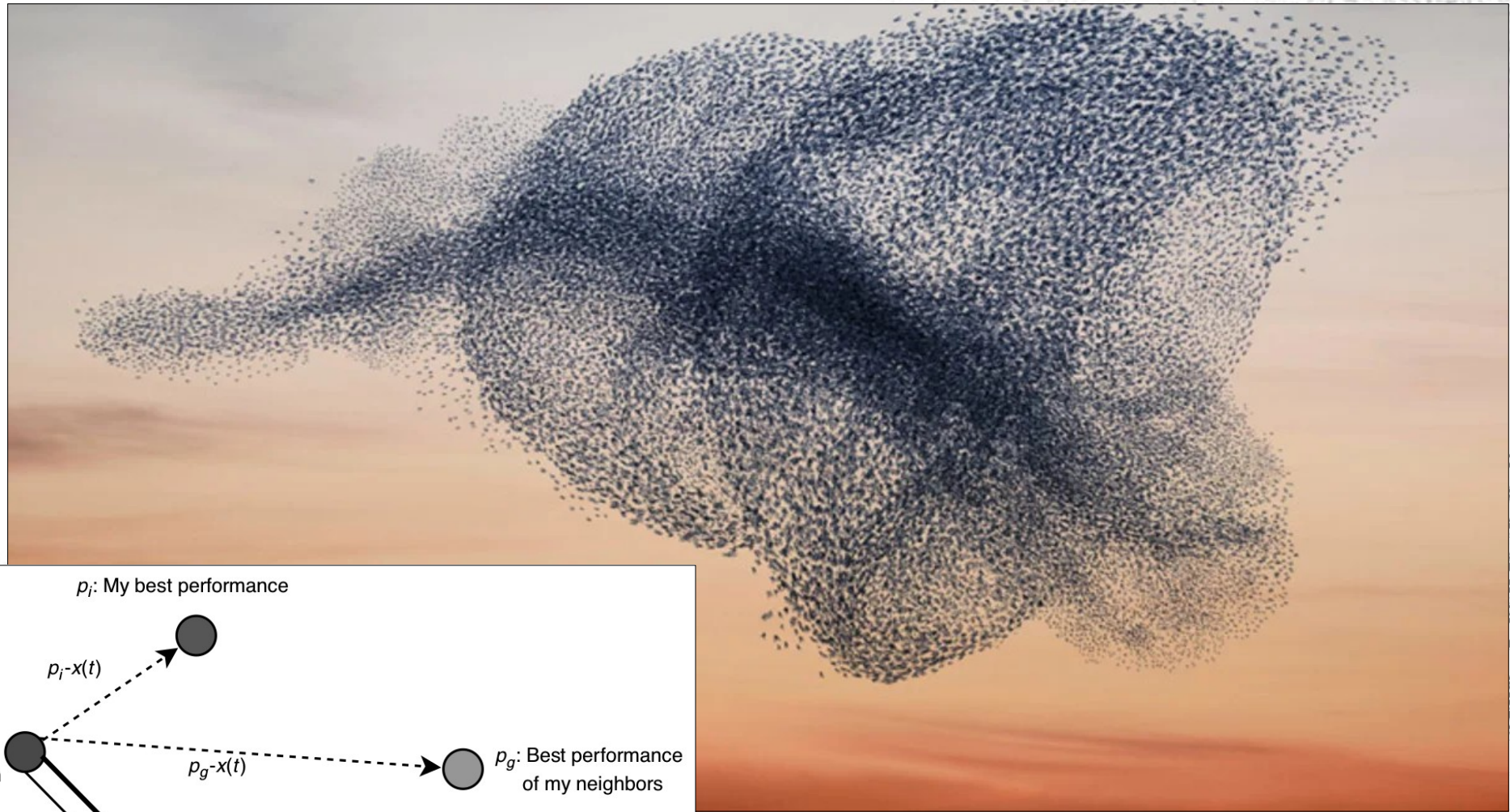
Necesidad de metaheurísticas



Metaheurísticas

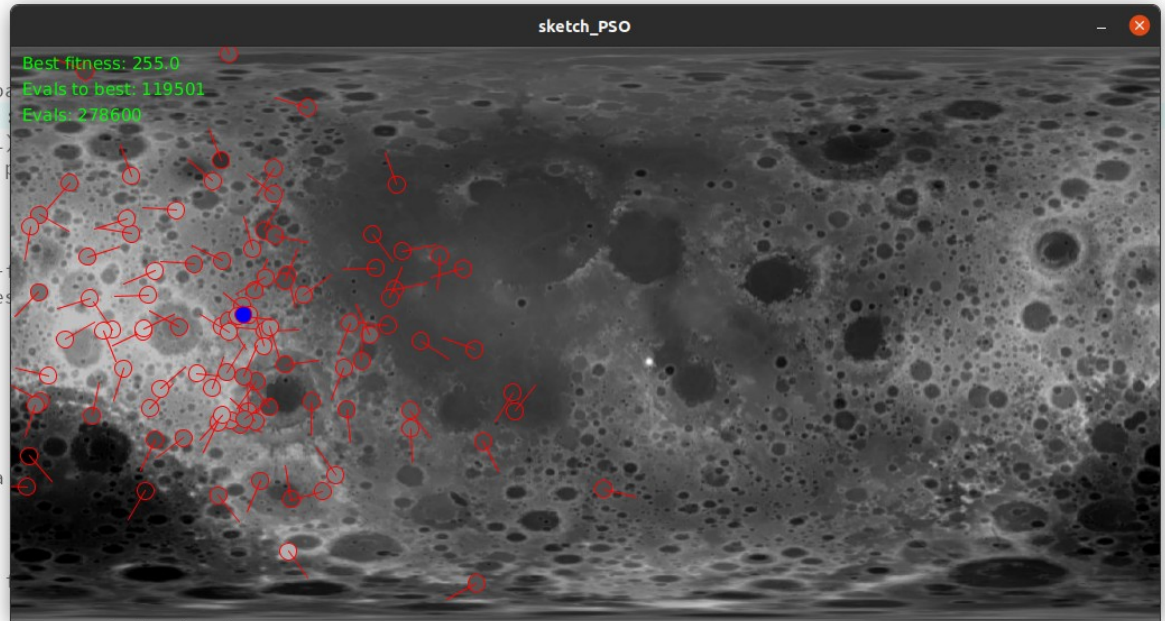


Particle Swarm Optimization (PSO)

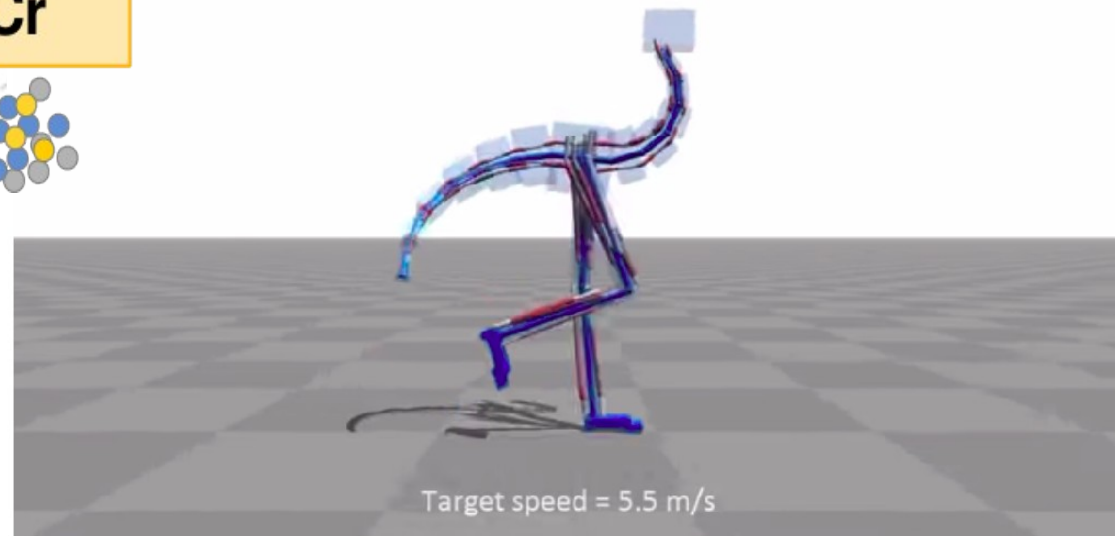
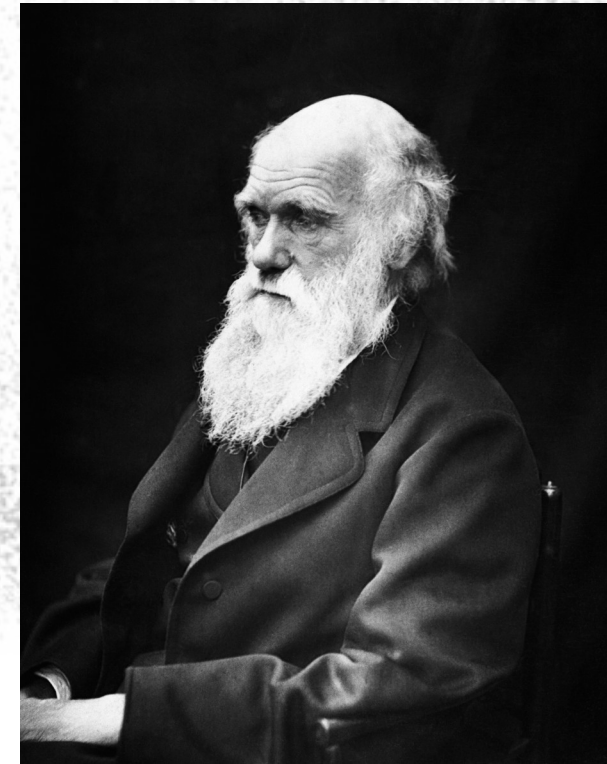
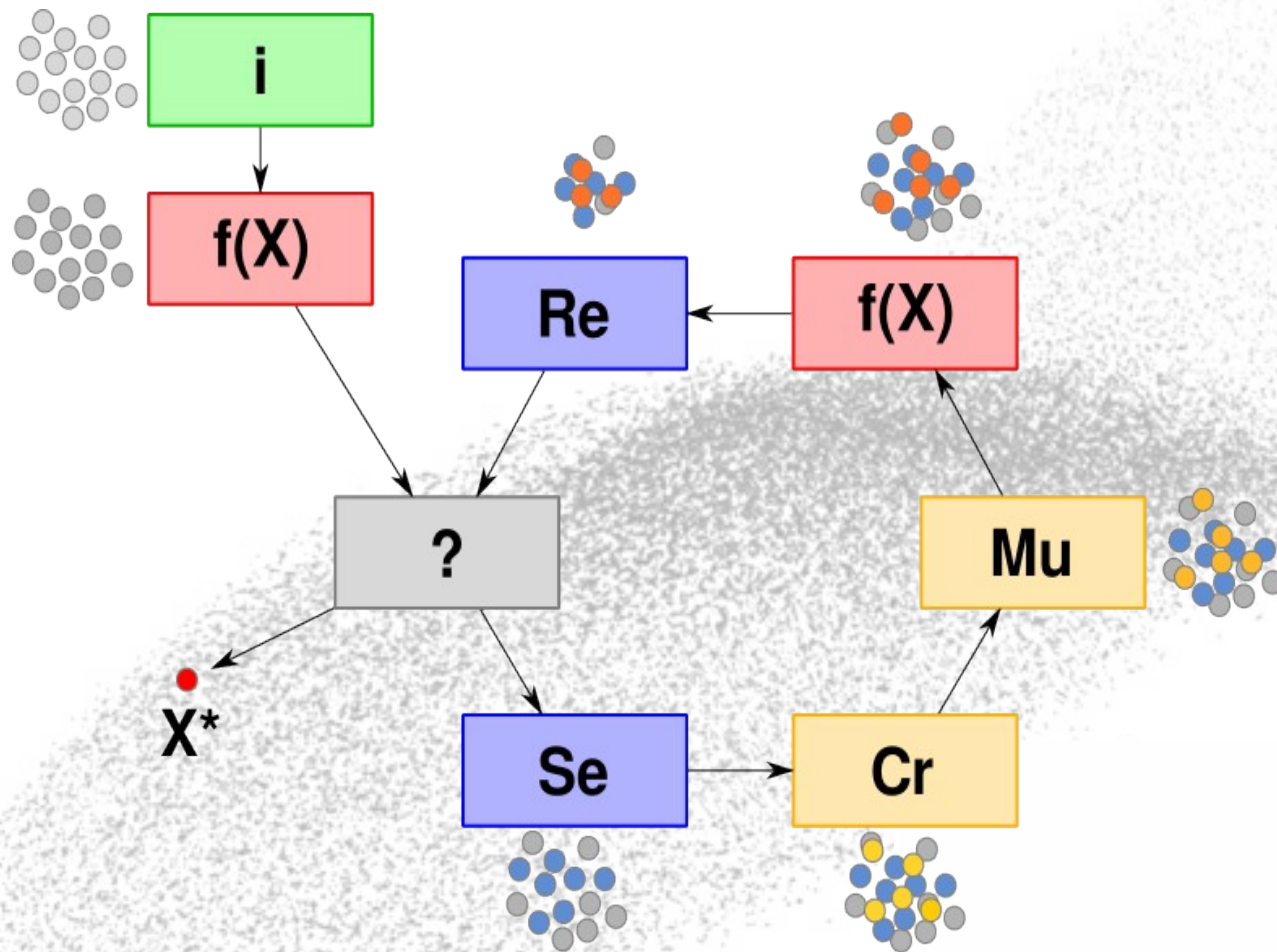


Implementación de PSO en Processing

```
sketch_PSO
1 // PSO de acuerdo a Talbi (p.247 ss)
2
3 PImage surf; // imagen que entrega el fitness
4
5 // =====
6 int puntos = 100;
7 Particle[] fl; // arreglo de partículas
8 float d = 15; // radio del círculo, solo para despliegue
9 float gbestx, gbesty, gbest; // posición y fitness del mejor global
10 float w = 1000; // inercia: baja (~50): explotación, alta (~5000): exploración
11 float C1 = 30, C2 = 10; // learning factors (C1: own, C2: social)
12 int evals = 0, evals_to_best = 0; // número de evaluaciones, sólo para global
13 float maxv = 3; // max velocidad (modulo)
14
15 class Particle{
16   float x, y, fit; // current position(x-vector) and fitness (x-vector)
17   float px, py, pfit; // position (p-vector) and fitness (p-vector)
18   float vx, vy; //vector de avance (v-vector)
19
20   // ----- Constructor
21   Particle(){
22     x = random(width); y = random(height);
23     vx = random(-1,1); vy = random(-1,1);
24     pfit = -1; fit = -1; //asumiendo que no hay valores menores a -1
25   }
26
27   // ----- Evalúa partícula
28   float Eval(PImage surf){ //recibe imagen que define función de fitness
29     evals++;
30     color c=surf.get(int(x),int(y)); // obtiene color de la imagen en posición (x,y)
31     fit = red(c); //evalúa por el valor de la componente roja de la imagen
32     if(fit > pfit){ // actualiza local best si es mejor
33       pfit = fit;
34     }
35   }
36 }
```



Algoritmos Evolutivos



Evaluación

- **Trabajo grupal (3 o 4 integrantes)**
 - Programar PSO y EA para un problema estándar
 - Generar reporte escrito + videos
- **Prueba individual**
 - Evaluar adquisición de resultados de aprendizaje
 - Prueba de desarrollo

Calendario y vías de comunicación

- Calendario

- **31 de julio**

- Clase introductoria (ésta)
 - Conformación de grupos

- **7 y 14 de agosto**

- Reuniones de seguimiento y consulta

- **21 de agosto**

- Entrega de informe
 - Prueba

- Comunicación

- **Slack** para mensajería
 - **SiveducMD** para publicación de contenidos, entrega de informe y toma de prueba