

Algorítmica II. Algoritmos Metaheurísticos

Joaquín Roiz Pagador¹

Universidad Pablo de Olavide, Carretera Utrera s/n, Sevilla, España,
quiniroiz@gmail.com

Abstract. El siguiente documento pertenece a una serie de documentos que pretende servir a modo de resúmen para el temario de Algorítmica II para el Grado en Ingeniería Informática en Sistemas de Información.

Keywords: algoritmos, metaheurística, búsqueda, tabú

Algoritmos Genéticos.

Los **algoritmos genéticos** o **evolutivos** parten de la idea de tomar diferentes poblaciones, es decir, conjuntos de individuos. Cada individuo será una solución a nuestro problema, con un *fitness* adjunto que se irá mejorando.

Se generará una población (orden, aleatoriedad). Posteriormente se obtendrá un *fitness* y evolucionaremos estas soluciones, las mejores para las siguientes generaciones se cruzarán de forma similar a la recombinación genética.

Referenciaremos "La ley del más fuerte", aumentando la **probabilidad** de adaptación y **reproducción** o **cruce**.

Evolución Natural.

- **Individuo** → Habilidad de Reproducirse.
- **Población** *rightarrow* Capaz de reproducirse.
- **Variedad** entre individuos.
- Diferencias en la **habilidad** de **sobrevivir** en entornos asociadas a la variedad.

La **evolución** es un proceso que opera sobre los **cromosomas** más que sobre las estructuras de la vida que están codificadas en ellos.

Selección natural *rightarrow* Enlace entre cromosomas y actuación de sus estructuras decodificadas.

En la reproducción actuará la evolución. La evolución biológica **no tiene memoria**.

Inteligencia Computacional o Soft-Computing.

- Redes Neuronales.
- Computación Evolutiva.
- Sistemas Difusos.

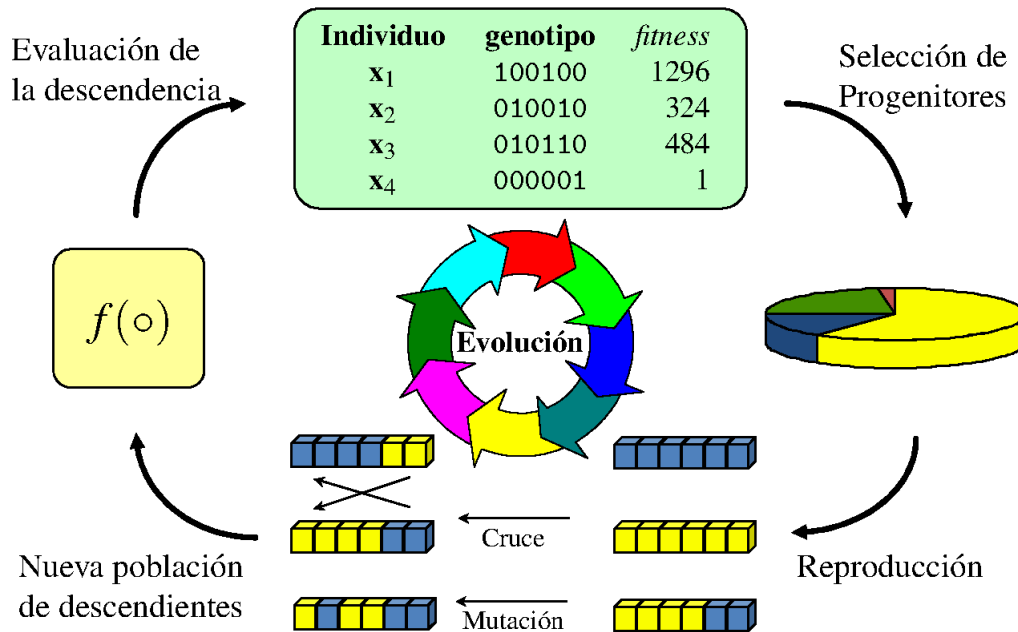


Fig. 1. Pasos para construir un Algoritmo Genético

```

Fun algEvo() : Sol is
    t = 0;
    inicializarP(t); ← Poblacionprimera
    evaluarP(t); ← evaluarpoblacionprimera
    while !condParada() do
        t = t + 1; ← generacion ++
        seleccionarP' desde P(t - 1) ← trabajoygeneronuevaymejorpoblacion.
        recombinarP'; ← combinonconlapoblacionanterior
        mutarP'; Si sedaelcaso, semutara
        reemplazarP(t) apartir de P(t - 1) y P'; ← evolucionala poblacion
        evaluarP(t); ←
            estudiarsufitnessgeneral(siel fenotipo es el deseado, puede ser criterio de parada)
    end
    return P(t);
end

```

Algorithm 1: Estructura de un algoritmo evolutivo.

Computación Evolutiva.

Basados en modelos de evolución con poblaciones, esta técnica de simulación para optimización probabilística que mejora los métodos clásicos en problemas difíciles.

Se buscará un equilibrio con **diversificación** e **intensificación**. Los Algoritmos Genéticos establecen búsqueda con equilibrio idóneo entre los anteriores.

1. Representación.

1	0	1	0	Representación binaria
1	2	3	4	Representación Secuencial

2. Inicialización → Elegir población a partir de **heurística previa**.

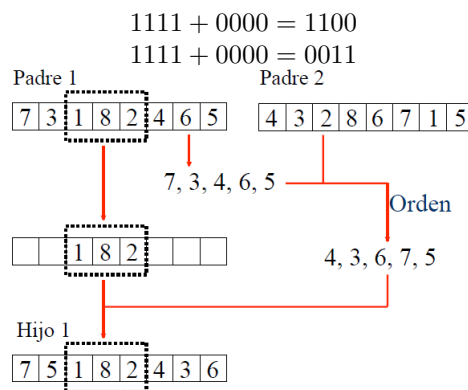
3. Genotipo y Fenotipo → Deberemos decidir el camino y conjunto de parámetros más adecuados.

4. Evaluación (Fitness) → **COSTOSO** → medimos la bondad (cuánto buena o mala es la solución).

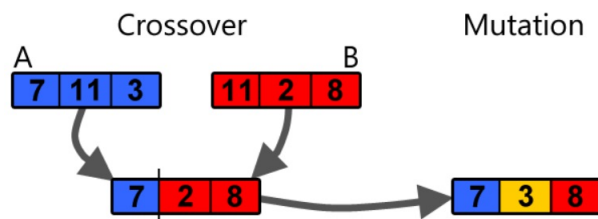
5. Mutación → Diversificación.

$$1111 \rightarrow 1101$$

6. Cruce → Intensificación.



7. Cruce y Mutación.



8. Estrategias de selección

- Los mejores deben tener más posibilidades frente a peores individuos.
- Cuidado para dar oportunidad a menos buenos.

(a) **Elitista** → Los miembros más aptos de cada generación.

(b) **Proporcional a aptitud** → Mejores tienen más probabilidad, pero no certeza.

(c) **Rueda de ruleta** → Probabilidad de selección proporcional a la diferencia entre su aptitud y la de sus competidores.

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

- (d) **Escalada** → Si aumenta la aptitud media de población, aumenta la presión selectiva y la función aptitud discrimina más.
- (e) **Torneo** → Subgrupos de individuos que compiten entre ellos. Gana uno de cada grupo.
- (f) **Por Rango** → A cada individuo se asigna un rango numérico basado en su aptitud. Selección basada en ranking, y no en diferencias absolutas en aptitud.
- (g) **Generacional** → La generación se basa en cada descendencia. No se conservan individuos entre las generaciones.
- (h) **Jerárquica** → Múltiples rondas de selección en cada generación. En primeros niveles más rápidas y menos discriminatorias. A mayor nivel, más rigurosas.

Ejemplo Torneo.

- Se escogen aleatoriamente k individuos.
- Seleccionar el mejor de ellos.
- k es el tamaño del torneo. A mayor k, mayor presión selectiva.

- 9. **Reemplazo** → Individuos que son reemplazados de la población.
- 10. **Criterio de parada.**
 - (a) Óptimo alcanzado
 - (b) Recursos limitados CPU (Iteraciones)
 - (c) Iteraciones sin mejora

Utilización de un GA.

- No sacar conclusiones de una única ejecución.
 - Usar Medidas estadísticas.
 - Realizar ejecuciones independientes.
- **2 Puntos de vista:**
 - Encontrar 1 solución muy buena al menos 1 vez.
 - Encontrar al menos solución muy buena en cada ejecución.
- Equilibrar Diversificación e Intensificación.

Inteligencia Colectiva o de Enjambre (Swarm Intelligence).

Similar comportamiento al conjunto de células para especializarse, formarse o realizar acciones específicas sin haber recibido orden de "un supervisor". Como ejemplo podemos observar un panel de abejas construido por el enjambre o un hormiguero construido por hormigas.

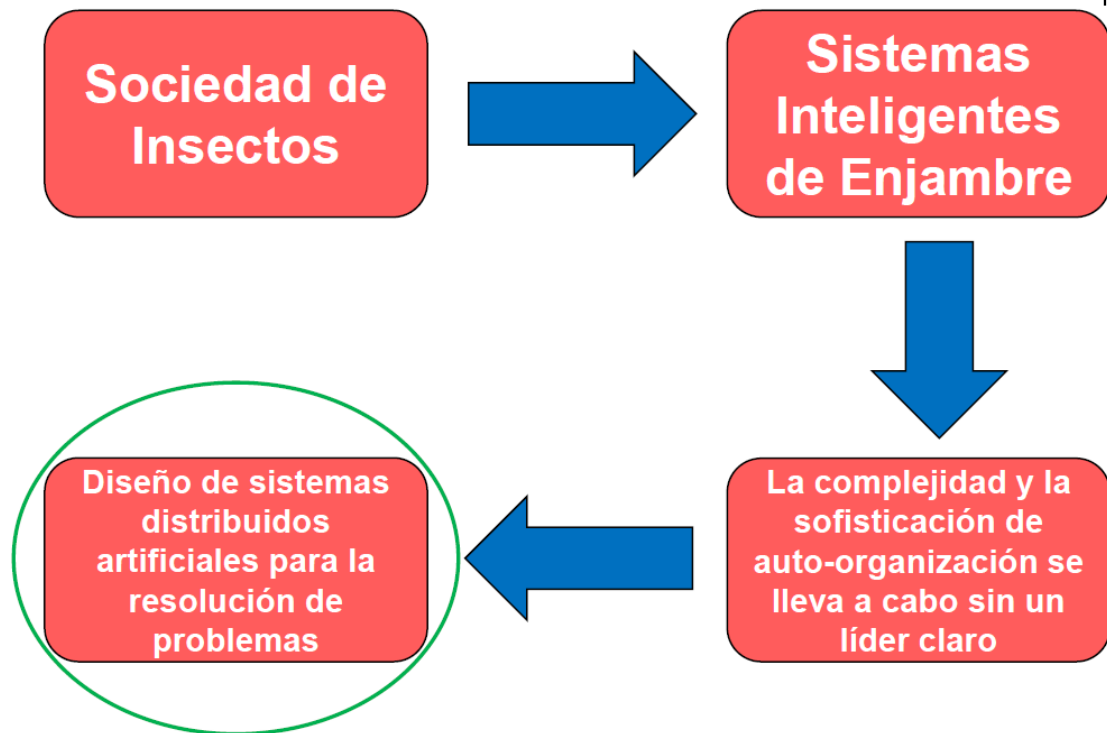
La inteligencia se basará en la población, no en el individuo.

Swarm Intelligence.

- Particle Swarm.
- Ant Colony.
- Bee Colony.

La *Inteligencia Colectiva* emerge de grupo de agentes simples. Son algoritmos **distribuidos** de resolución inspirados en el comportamiento colectivo de colonias de insectos sociales u otras especies de animales.

Los individuos simples colectivamente configurados como un enjambre producen resultados inteligentes: Inteligencia Colectiva.



Comportamiento Emergente.

Reglas simples y comunicación local simple.

- Abejas.
 - Cooperación Colmena.
 - Regulan temperatura.
 - Eficiencia vía especialización. División de la labor.
 - Comunicación: Fuentes de comida explotadas de acuerdo a la calidad y distancia desde la colmena(bailes).
- Termitas.
 - Nidos.
 - Cámaras de camadas.
 - Rejillas ventilación.
 - Columnas soporte.
- Hormigas.
 - Autopistas hacia y desde la comida a través de feromonas.

Inspiración Biológica: Enjambre.

Un enjambre se compone de agentes simples y auto-organizados (Self-Organized). A su vez, es descentralizado, es decir, no hay un único supervisor. No hay un plan global (emergente). Es robusto: las actuaciones se completan aunque un individuo falle. También es flexible:

1. Puede responder a cambios externos.
2. Percepción del entorno (sentidos).
3. No existe un modelo explícito de entorno para cambiarlo.

Ideas Clave. La complejidad y sofisticación de la auto-organización se lleva a cabo sin un líder/jefe de la sociedad. Según aprendemos de los insectos sociales, se puede aplicar al campo del diseño de *Sistemas Inteligentes*. La modelización de los insectos sociales por medio de la auto-organización puede ser de ayuda para el diseño de modelos artificiales distribuidos de resolución de problemas. Esto es conocido como **Swarm Intelligent Systems**.

Técnicas.

- **Particle Swarm Optimization(PSO)**: Optimización basada en nubes de partículas.
 - Conjunto de técnicas inspiradas en el comportamiento de las bandadas de aves o bancos de peces. Aplica conceptos de interacción social a la resolución de problemas de búsqueda/optimización.
 - En PSO, un enjambre de n individuos se comunica directa o indirectamente con otros vía las direcciones de búsqueda.
 - Las partículas pueden ser simples agentes que vuelan a través del espacio de búsqueda y almacenan (y posiblemente comunican) la mejor solución que han descubierto.
 - En PSO, las partículas nunca mueren.
- **Ant colony optimization(ACO)**: Optimización basada en colonias de hormigas.
 - Conjunto de técnicas inspiradas por las actividades de una colonia de hormigas. La analogía más cercana a **ACO** son los problemas de ruta en grafos:
 - * Mientras las hormigas buscan comida, depositan rastros de feromona que atraen a otras hormigas.
 - * Desarrollan caminos mínimos entre la comida y el hormiguero.
- **Bee Colony Optimization(BCO)**: Optimización basada en colonias de abejas.
 - Conjunto de técnicas inspiradas por las actividades de una colonia de abejas.
 - Los algoritmos de optimización basados en abejas están basados en la recogida de comida (néctar) de las abejas.

Aplicaciones.

- Búsqueda de Rutas para una Red de Satélites:
 1. Di Caro, Dorigo y otros autores mostraron que **ACO** da buenos resultados en la búsqueda de rutas en grandes sistemas de telecomunicaciones y redes de ordenadores.
 2. **AntNet** es un algoritmo para la adaptación de rutas en redes y ha sido diseñado basado en *Ant Colony Optimization* (ACO).
- Una red utilizada en el estudio. La red de datos de Japón: **NNTnet**.
- Interacciones animal-robot.

References

1. Material asignatura, *Metaheurísticas*, UGR. <http://sci2s.ugr.es/graduateCourses/Metaheuristics>
2. E.-G. Talbi, *Metaheuristics. From design to implementation*, Wiley, 2009 Material.
3. C. Blum, A Roli, *Metaheuristics in Combinatorial Optimization: overview and conceptual comparison*. *ACM Computing Surveys*, 35(3), 2003, 268-308.
4. B. Melián, F. Glover. *Introducción a la búsqueda tabú*, 2006. [http://leeds-faculty.colorado.edu/glover/fred%20pubs/329%20-%20Introduccion%20a%20la%20Busqueda%20Tabu%20TS_Spanish%20w%20Belen\(11-9-06\).pdf](http://leeds-faculty.colorado.edu/glover/fred%20pubs/329%20-%20Introduccion%20a%20la%20Busqueda%20Tabu%20TS_Spanish%20w%20Belen(11-9-06).pdf)