



Algoritmos Bioinspirados

Unidad I : Optimización

Programa: Ingeniería en Inteligencia Artificial

Optimización

En general, optimizar significa buscar la solución a un problema de la mejor manera, esta búsqueda puede estar sujeta a restricciones.

Muchos de los problemas a los que nos enfrentamos requieren optimización.

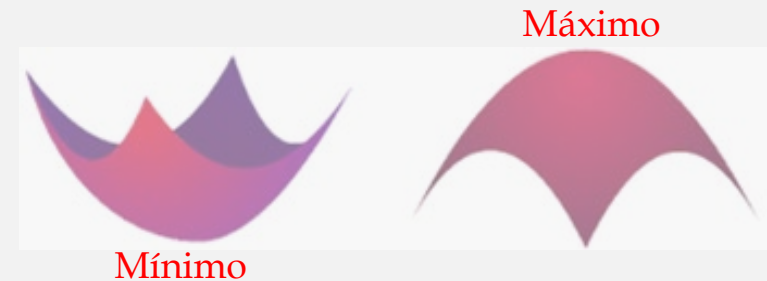
Optimización – Ejemplos

- Programación de vuelos para aerolíneas.
- Planificación de rutas de entrega
- Optimización de la producción de una fábrica para maximizar ganancias



Optimización – Objetivo

Maximizar o minimizar una función



Optimización – Definición

Consideramos el problema de optimización *minimizar o maximizar* $f(x)$ donde x es un vector n -dimensional de variables independientes

$$f(x) = (x_1, x_2, \dots, x_n) \in R^n$$

Sujeto a restricciones

$$h(x) = 0$$

$$g(x) \leq 0$$

Optimización – Ejemplo

Imaginemos un problema donde llenar una mochila con una capacidad determinada con un conjunto de objetos, que tienen un peso y un valor, donde los objetos colocados en la mochila deben maximizar el valor sin exceder el peso.

Capacidad
40 Kg.



Peso: 5 Kg
Ganancia: 50



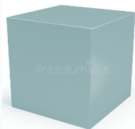
Peso: 10 Kg
Ganancia: 70



Peso: 20 Kg.
Ganancia : 60



Peso: 15 Kg.
Ganancia: 100



Peso = 15; ganancia = 100

Peso = $(20+15+10+5) > 40$

Peso = $20+10+5$; ganancia = $60+70+50$

Optimización – Ejemplo

Las redes neuronales son también un problema de optimización

$$\text{Minimizar } E(w) = \frac{1}{n} \sum_{i=1}^n [u_i(x_i, w) - \bar{y}_i]^2$$

Cuando el espacio de soluciones S es pequeño, como en el ejemplo anterior, se puede recurrir a una búsqueda exhaustiva. Lo cual significa generar todas las soluciones posibles, evaluarlas y por tanto encontrar la mejor.

El problema es que en la mayoría de los problemas reales S es demasiado grande para permitir una búsqueda exhaustiva.

En problemas reales el espacio de soluciones puede ser muy grande por lo que los problemas de optimización pueden ser difíciles de resolver. Por ejemplo, los problemas combinatorios tienden a ser muy difíciles (NP-Hard). Donde una búsqueda exhaustiva no es posible.





Optimización – Terminología

Función objetivo: la función f que se debe minimizar o maximizar.

Variables de diseño: el vector $x = (x_1, x_2, \dots, x_n) \in R^n$ son los parámetros que el algoritmo de optimización puede elegir para minimizar la función objetivo. Estas variables deberían ser independientes entre si.

Restricciones: son funciones sobre las variables de diseño que se quieren restringir. Cuando la función debe ser igual a un valor es una restricción de igualdad. Cuando la función debe ser mayor o igual a un valor es una restricción de desigualdad

Planteamiento de problema de optimización

$$\begin{array}{ll} \min/\max & f(x) \\ \text{variando} & x \in R^n \\ \text{sujeto a} & h_j(x) = 0, \quad j = 1, 2, \dots, m \\ & g_k(x) \leq 0, \quad k = 1, 2, \dots, l \end{array}$$

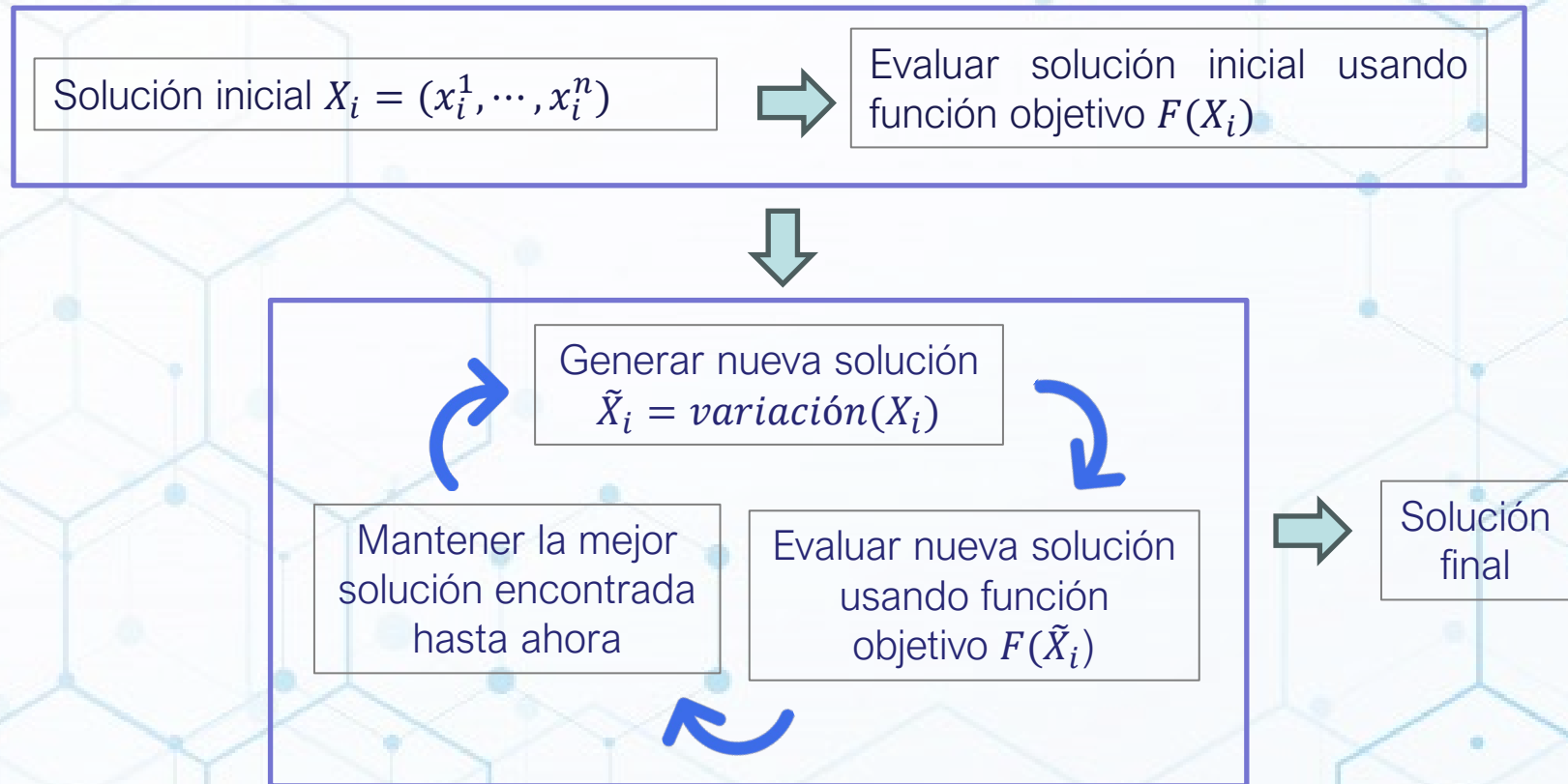
f : función objetivo, salida.

x : vector de variables de diseño, entradas.

h : vector de restricciones de igualdad

g : vector de restricciones de desigualdad

Un algoritmo de optimización es un procedimiento iterativo, que comienza con una estimación inicial. Después de un cierto número (suficientemente grande) de iteraciones, puede converger hacia una solución estable, idealmente la solución óptima a un problema de interés



Definición

- ❖ En optimización un espacio de búsqueda es el dominio de la función que se busca optimizar.
- ❖ Espacio de soluciones, donde se encuentran las soluciones candidatas.

Proceso de búsqueda

Una búsqueda sobre un espacio consiste en generar una sucesión de puntos del espacio en el que cada punto se obtiene del anterior por medio de una serie de transformaciones o movimientos.

En problema de optimización se realizan recorridos sobre el espacio de posibles soluciones y se selecciona la mejor solución encontrada en el recorrido.

Búsqueda Local

Una búsqueda local es un proceso que, dada la solución actual en la que se encuentra el recorrido, selecciona iterativamente una solución de su vecindad inmediata para continuar la búsqueda.

Ventaja

Es un enfoque rápido para encontrar soluciones buenas.

Desventaja

Puede quedarse atrapada en un óptimo local, que es una solución mejor que su entorno, pero no la mejor solución global.

Búsqueda global

Un proceso que busca la mejor solución posible en todo el espacio de soluciones de un problema. Explora todo el espacio de soluciones, en lugar de solo la vecindad de la solución actual.

Las búsquedas globales incorporan mecanismos como:

- ❖ volver a comenzar la búsqueda desde otra solución inicial,
- ❖ permitir movimientos de empeoramiento de la solución actual

Ventaja

Tiene una mayor probabilidad de encontrar la solución óptima global, incluso en problemas con múltiples óptimos locales.

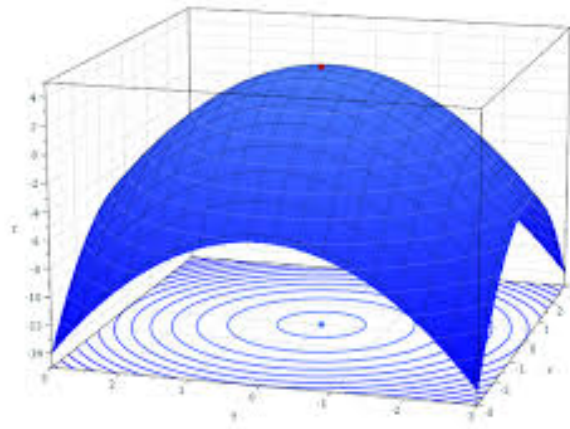
Desventaja

Generalmente es más costosa computacionalmente.

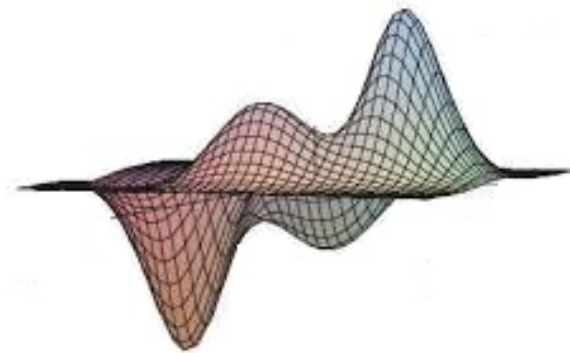


Si se conoce el espacio de búsqueda se puede seleccionar el tipo de algoritmo adecuado.

Sin embargo, esta situación raras veces se presenta por lo que se deben preferir métodos que realicen búsquedas globales. **Siempre que el costo computacional lo permita.**



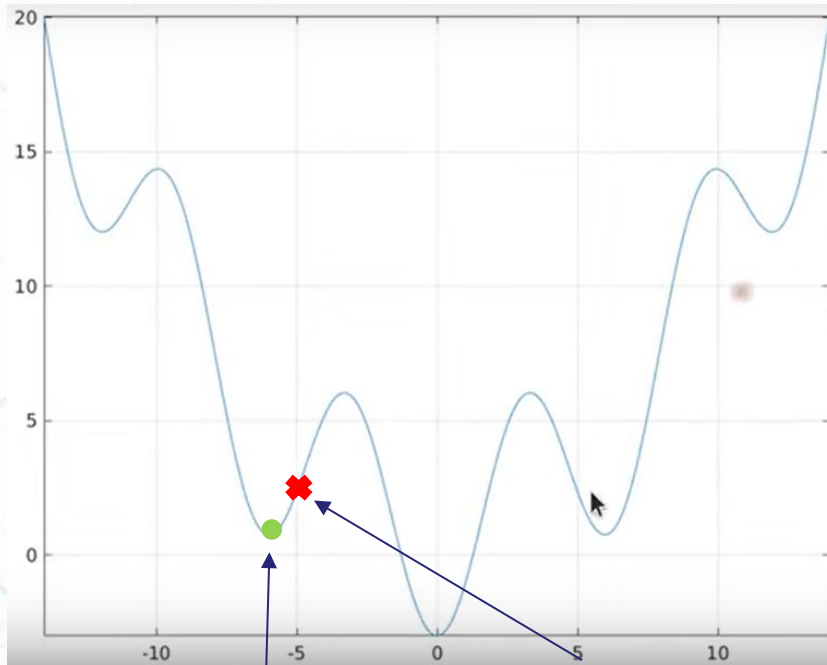
Espacio de búsqueda
con máximo único



Espacio de búsqueda con
múltiples mínimos y
máximos locales

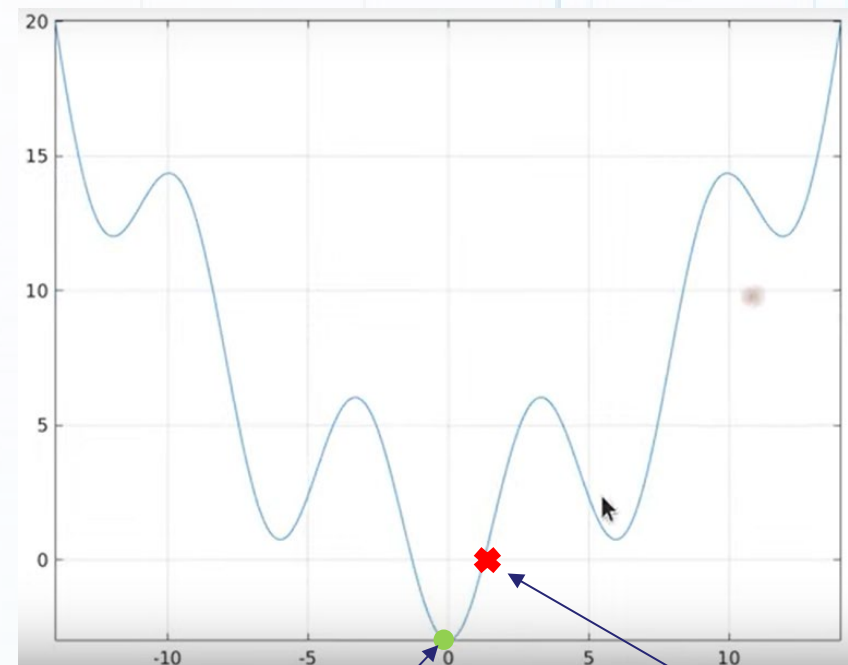
Minimizar la función $f(x) = 0.1x^2 - 4 \cos(x) + 1$

Ejemplo de diferencias debido al valor inicial de la variable



Punto inicial $x = -5$

Óptimo encontrado
 $x = -5.98$



Punto inicial $x = 1$

Óptimo encontrado
 $x = 0$

Tipos de algoritmos de optimización

- ❖ **Programación lineal:** se centra en maximizar o minimizar funciones lineales sujetas a restricciones lineales. Es ampliamente utilizada en la planificación de recursos y la optimización de cadenas de suministro.
- ❖ **Búsqueda local:** este enfoque mejora una solución inicial al explorar iterativamente su entorno cercano, lo que es ideal para problemas con múltiples óptimos locales.
- ❖ **Algoritmos genéticos:** inspirados en la evolución biológica, estos algoritmos simulan procesos como la selección natural, mutación y cruce para encontrar soluciones óptimas en problemas complejos.
- ❖ **Enjambre de partículas:** basado en el comportamiento colectivo de grupos de agentes, este algoritmo encuentra soluciones óptimas al permitir que los agentes compartan información y ajusten sus posiciones en el espacio de soluciones.



Técnicas de Optimización - Tradicionales

- ❖ Programación lineal
- ❖ Programación cuadrática
- ❖ Descenso de gradiente
- ❖ Búsqueda local

La mayoría de los problemas reales no son lineales o convexos, por lo que estos métodos a menudo les cuesta encontrar una solución o simplemente no funcionan.

Técnicas de Optimización – Nuevas tendencias

- ❖ Algoritmos evolutivos (algoritmos genéticos, programación genética).
- ❖ Inteligencia de enjambres (colonia de hormigas, enjambre de partículas, ...).

Algoritmos de optimización inspirados en la naturaleza basados en procesos estocásticos y en poblaciones.

Optimización – Desventajas de enfoque tradicional

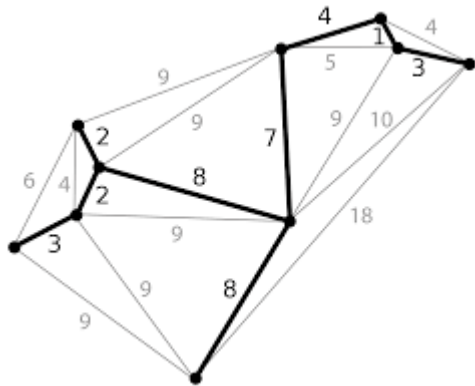
- ❖ Generalmente se basan en búsquedas locales, por lo que no garantiza un óptimo global.
- ❖ Resultados depende de punto inicial de búsqueda.
- ❖ Enfrentan dificultades en problemas no continuos.

Optimización – Algoritmos bioinspirados

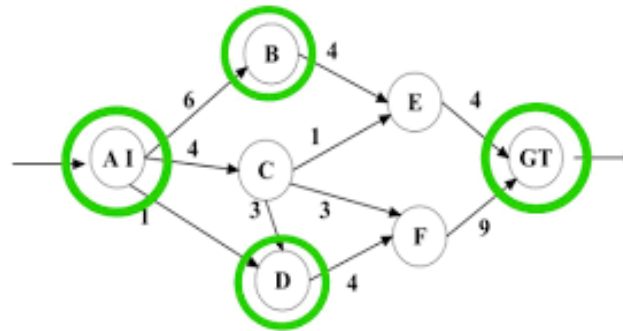
- ❖ Incrementan la posibilidad de encontrar un óptimo global.
- ❖ Pueden ser aplicados a un espectro más amplio de problemas.
- ❖ Pero pueden ser computacionalmente más costosos.

Problemas combinatorios

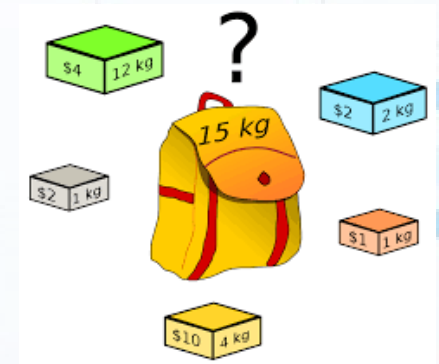
La principal característica de estos problemas es que la complejidad crece de forma exponencial con respecto al número de variables involucradas.



Problema árbol mínimo
(minimum spanning tree)



Problema flujo máximo



Problema de la mochila

Generalmente son problemas muy difíciles pues el número de candidatos a solución, aunque finito, es muy grande.

La estrategia más simple es la fuerza bruta.

Existen mejoras a esta aproximación como son las estrategias voraces y de ramificaciones.

Algunos algoritmos usados para resolver este tipo de problemas:

- ❖ Búsqueda local
- ❖ Ascenso de colina (Hill climbing)
- ❖ Recocido simulado (Simulated annealing)
- ❖ GRASP (Greedy Randomized Adaptive Search)
- ❖ Búsqueda Tabú



Algoritmos bioinspirados

Replican la forma en que la naturaleza se enfrenta a los problemas. Los principales paradigmas tratan de emular la evolución natural de las especies y la respuesta de sistemas sociales.

Predicción de mercados financieros



Diseño de estructuras



Diseño de rutas de escape y tácticas militares

Ventajas/Desventajas

Bastante genéricos y con muy pocas restricciones de uso.

Ofrecen poblaciones de soluciones.

Tienen naturaleza aleatoria.

Usualmente dependen de muchos parámetros.

Suelen tener problemas de convergencia prematura.



Principales elementos de la computación evolutiva

Esquema de representación: definir como se representan el conjunto de soluciones que forman el espacio de búsqueda para el algoritmo.

Población inicial: un conjunto inicial de soluciones creadas.

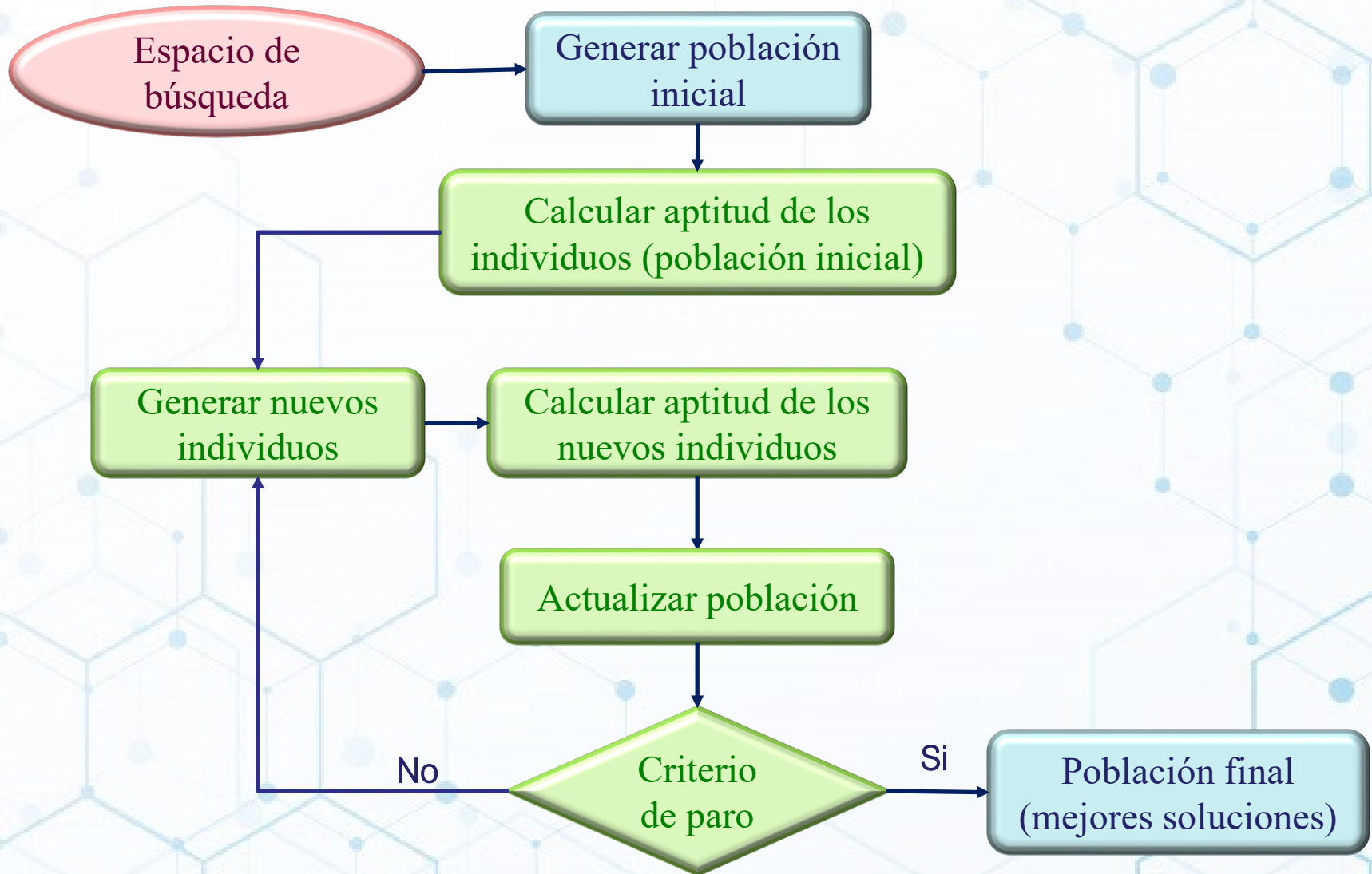
Criterio de paro: condición predefinida que controla si se alcanza una solución aceptable.

Función de aptitud: cada individuo de la población debe ser evaluado usando esta función.

Operadores de reproducción: son usados para la creación de nuevos individuos a partir de individuos ya existentes en la población.

Método de selección: el mecanismo que determina cuál y cuántos padres seleccionar, cuántos descendientes crear y qué individuos sobrevivirán en la próxima generación.

Diagrama general de un algoritmo genético



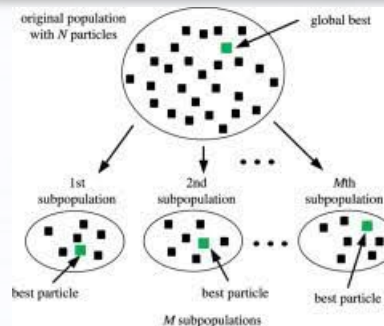
Inteligencia de enjambre

Estudia cómo obtener un comportamiento complejo a partir de la cooperación de agentes individuales con comportamientos relativamente simple.

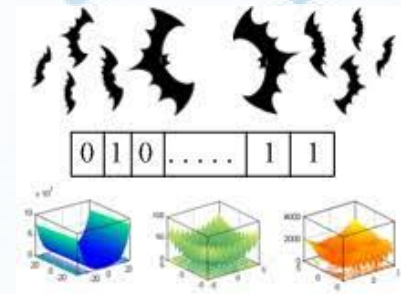
Diseño de algoritmos inspirados por el comportamiento colectivo en colonias de insectos y otros animales sociales.



Colonias de hormigas



Enjambre de partículas



Algoritmo de murciélago

- ❖ Algoritmo colonia de abejas artificiales
- ❖ Sistemas inmunológicos artificiales
- ❖ Algoritmo de búsqueda de cuco ...

Principales elementos de la inteligencia de enjambre

Población: conjunto de agentes simples que interactúan localmente entre ellos y su entorno.

Descentralizado: las interacciones entre los agentes tienen como resultado el comportamiento global del sistema. Son sistemas distribuidos y autoorganizados.

Comunicación: de forma directa con sus vecinos y de forma indirecta a través de señales en el entorno (estigmergia).



Problemas NP – Nociones Básicas

Tamaño del problema: el número de variables y todos los posibles valores que pueden tomar estas variables

Tiempo de ejecución: número de operaciones que le toma a un algoritmo le toma terminar la ejecución.

- ❖ Función de tamaño del problema en el peor caso
- ❖ Polinomial o exponencial

Problemas NP – Nociones Básicas

Clase P: el algoritmo puede resolver el problema en tiempo polinomial (el tiempo de ejecución en el peor de los casos para el tamaño del problema n es menor que $F(n)$ para alguna fórmula polinomial F).

Clase NP: el problema se puede resolver y cualquier solución se puede verificar dentro del tiempo polinomial mediante algún otro algoritmo (P subconjunto de NP).

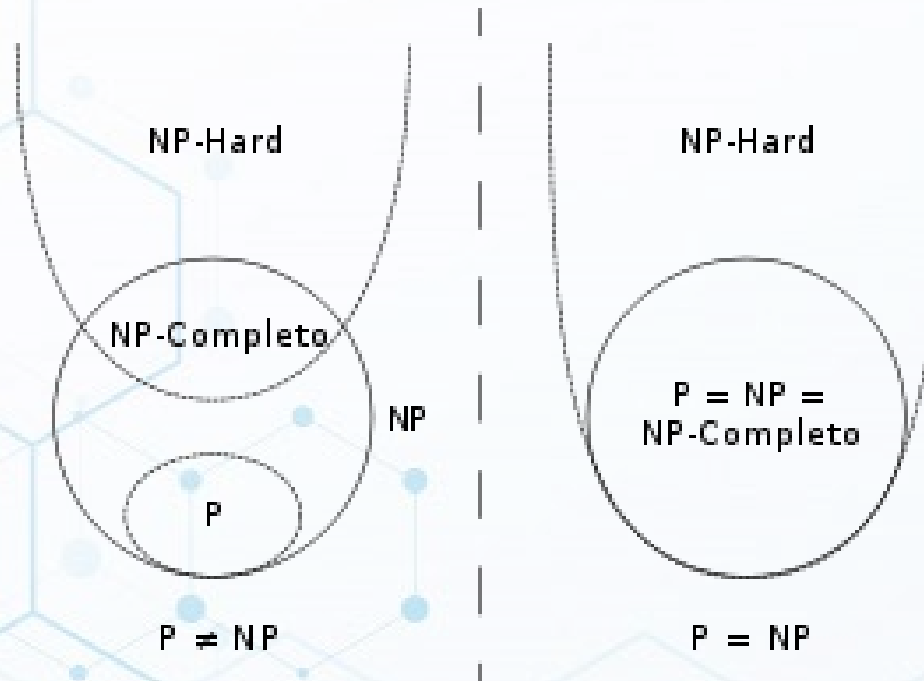
Clase NP-hard: cualquier otro problema en NP puede reducirse a este problema mediante un algoritmo que se ejecuta en tiempo polinomial, pero la solución no necesariamente se puede verificar dentro del tiempo polinomial

Clase NP-completo: el problema pertenece a la clase NP y cualquier otro problema en NP puede reducirse a este problema mediante un algoritmo que se ejecuta en tiempo polinomial.



Problemas NP – Nociones Básicas

- ❖ P es diferente de NP-hard.
- ❖ Aún no se demuestra si $P \neq NP$





¡ Gracias !

Thanks !

Obrigado

Xie xie ni

Domo arigatou

Спасибо

Merci

Grazie

Alfa Beta