

# Optimización

## *Computación Evolutiva*

M. E. Mazzei

UNA

# Conceptos

Optimización de una función: Consiste hallar el mejor valor de una función sobre un conjunto de alternativas. En general, se persigue hallar el máximo valor ( maximización) o el valor mínimo ( minimización). En términos matemáticos se plantea de la siguiente manera:

Minimizar  $f(\mathbf{x})$  en donde  $\mathbf{x} \in \mathcal{S}, \mathbf{x} \in \mathbb{R}^n$

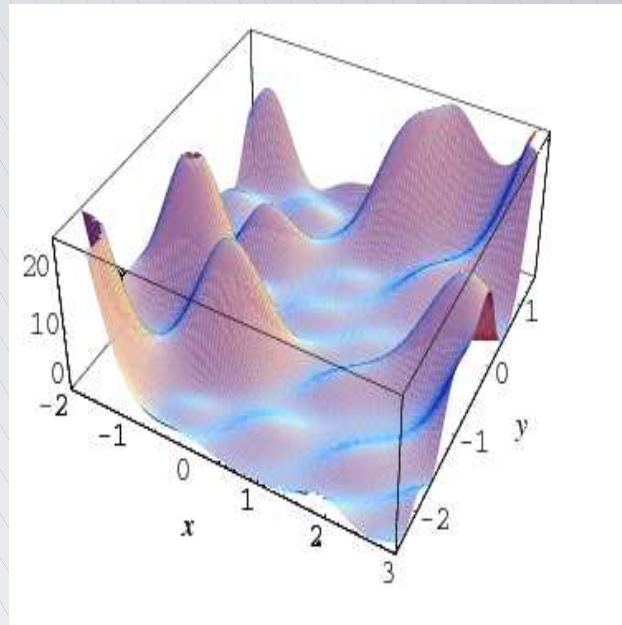
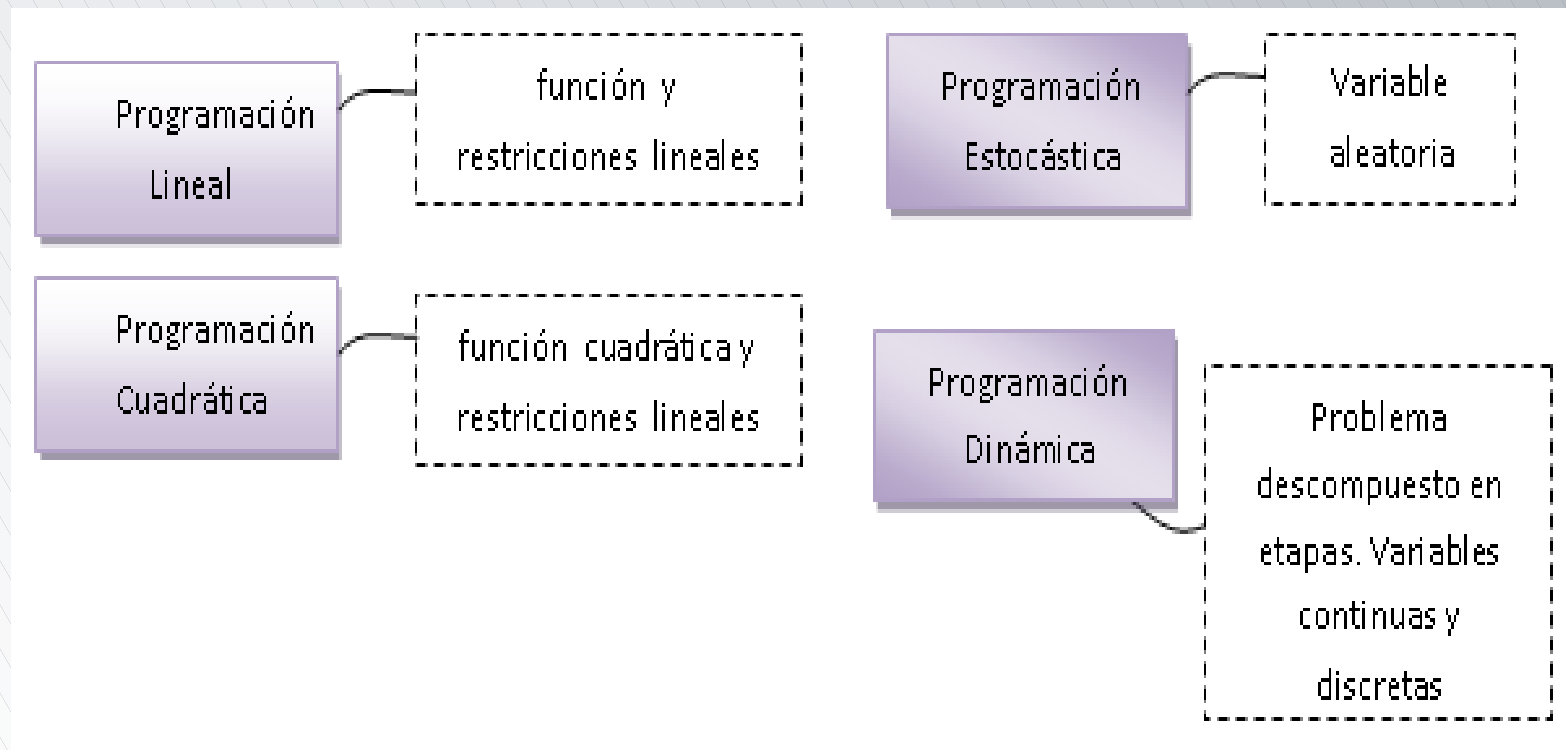


Gráfico de una función a optimizar. Fuente: [mathworld.wolfram.com](http://mathworld.wolfram.com)

# Taxonomía

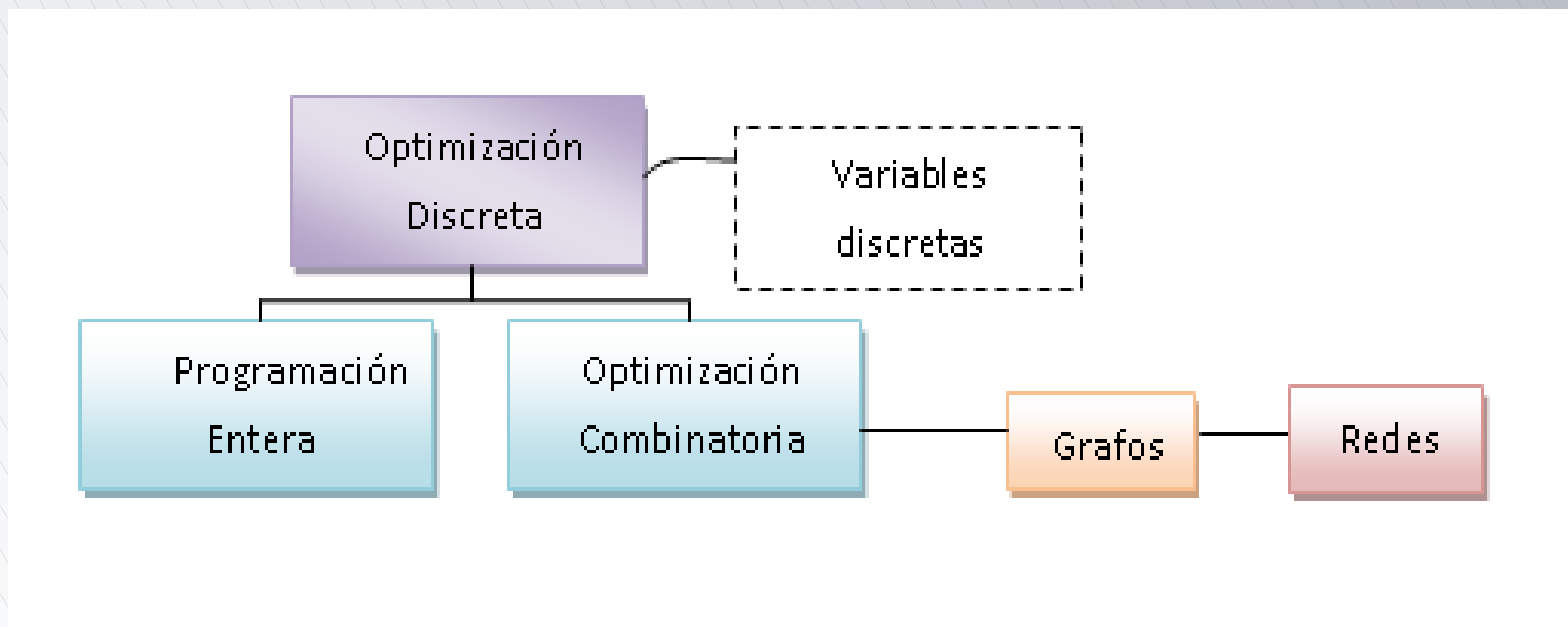
Existen muchas clasificaciones de los problemas de optimización, unas tienen que ver con el tipo de función y con su dominio, otros con el tipo de técnica empleada. El siguiente cuadro presenta algunos tipos:



Continúa...

# Taxonomía

Según algunos autores la optimización discreta comprende la programación entera y la combinatoria. El área de combinatoria involucra problemas de flujo de redes, camino más corto, mínimo árbol expandido, de coloración, de *matching*, de empacamiento y cubrimiento, entre otros.

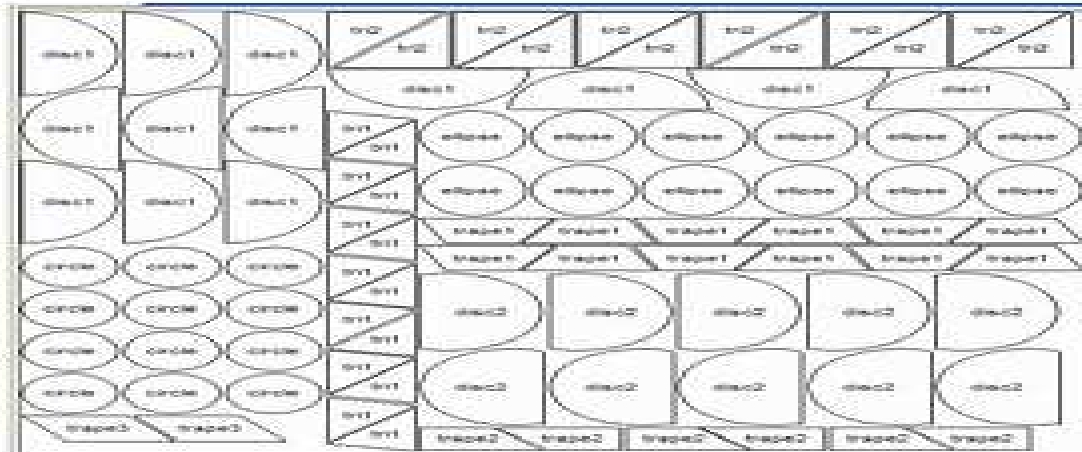


# Problema

Sea el problema del corte de patrones: Se tiene un conjunto de  $m$  órdenes de corte, cada una de las cuales debe constar de  $q_j$  piezas ( $j = 1, 2, \dots, m$ ). Si construimos una lista de todas las posibles combinaciones de cortes, denominados patrones.

En general, el número posible de patrones crece exponencialmente, como función del número de órdenes  $m$ , de tal manera que es poco práctico enumerar los posibles patrones de corte. **¿Cómo se resuelven este tipo de problemas?**

## Patrones de Corte



# Desde las técnicas clásicas hacia las Heurísticas

En computación existen algunos problemas que no tienen solución algorítmica, o que los algoritmos aplicables resultan muy costosos y por ello se requiere emplear ciertas técnicas para hallar su solución.

Para abordar la resolución existen muchos métodos clásicos y otros más recientes, entre ellos:

- **Métodos analíticos.** Si la función es de una variable y tiene dos derivadas en su rango, se pueden hallar sus mínimos(máximos). Pero a veces las funciones no son diferenciables.
- **Métodos exhaustivos:** Recorren todo el espacio de búsqueda, por lo tanto son costosos.
- **Métodos Aleatorios:** Se muestrea el espacio de búsqueda, se escoge la mejor solución y se calcula el intervalo de confianza de la solución hallada.
- Métodos que emplean **heurísticas:** Utilizan reglas para eliminar del espacio de búsqueda “zonas poco interesantes”.

Las técnicas empleadas para obtener la solución óptima a los problemas de optimización se han diversificado desde el tratamiento clásico de búsqueda local, pasando por las técnicas basadas en heurísticas, hasta la aplicación de algoritmos que emulan poblaciones, en el sentido biológico y aspectos relacionados con la genética como cruces y mutaciones.

Estos algoritmos basados en poblaciones también utilizan heurísticas.

# Heurísticas

En estos algoritmos se emplean heurísticas.

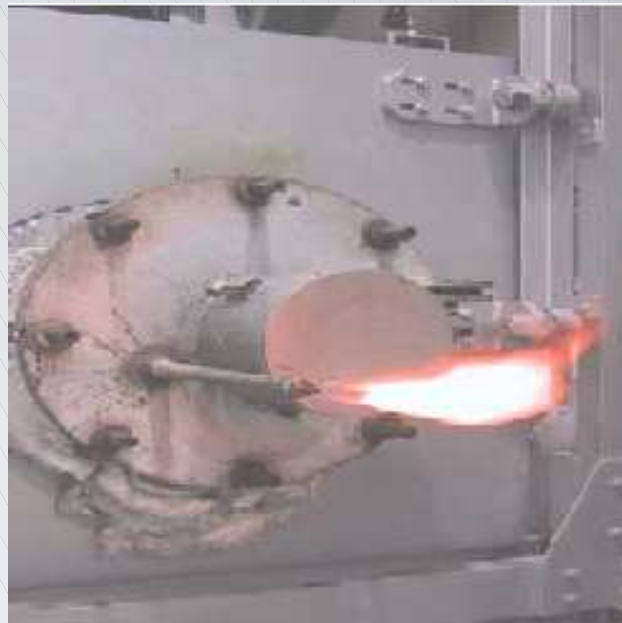
**Qué es heurística?:** La palabra “heurística” se deriva del griego *heuriskein*, que significa “encontrar” o “descubrir”. En el ámbito de la computación, las heurísticas se aplican para hallar buenas soluciones a un costo bajo, aunque no se garantice que éstas sean óptimas.

Son ejemplos de técnicas heurísticas: El Recocido Simulado, la Búsqueda Tabú y Escalando Colinas Estocástico, entre otras.



# Recocido Simulado ( *Simulated Annealing, SA* )

Es un método basado en un procedimiento físico empleado para mejorar las características de los materiales. El proceso consiste en calentar un material a temperaturas muy altas, hasta fundirlo. En esa situación, los átomos adquieren una distribución aleatoria dentro de la estructura del material. Se hace descender la temperatura muy lentamente por etapas, procurando que en cada una de esas etapas los átomos queden en equilibrio.



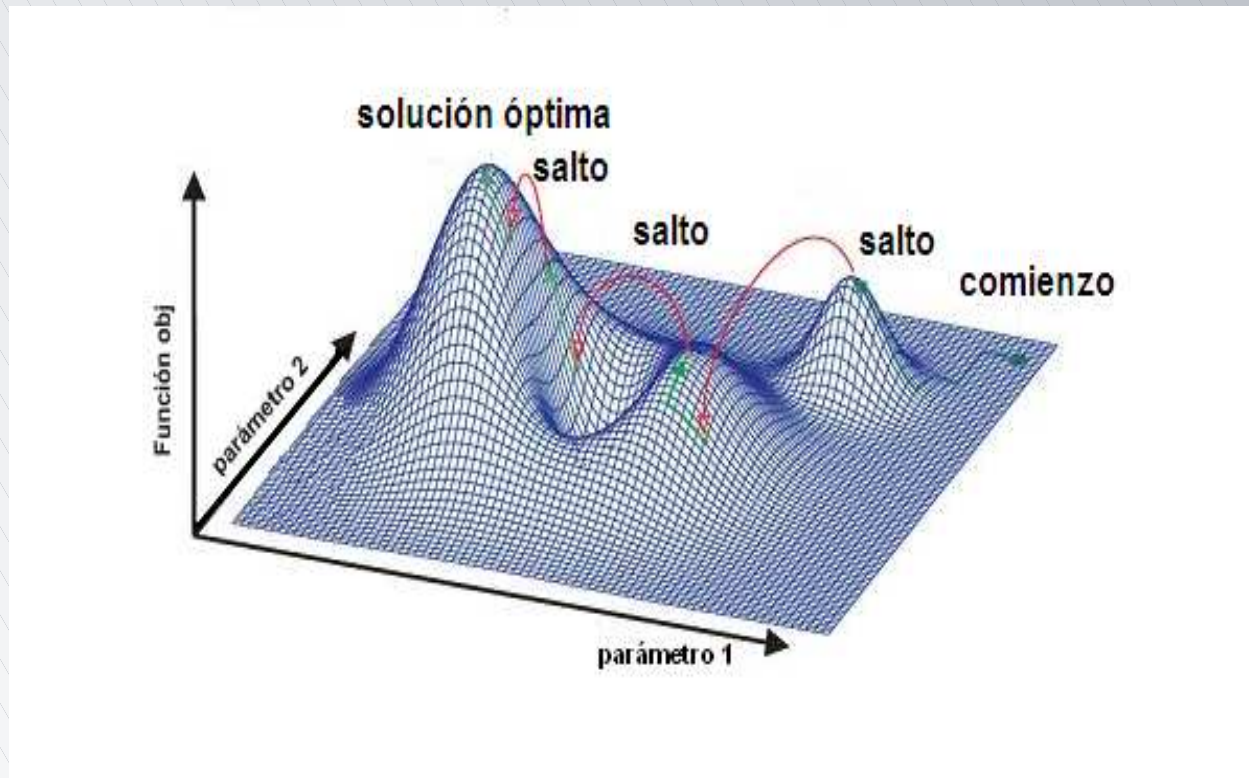
Fuente: [makano-permalloy.co.jp](http://makano-permalloy.co.jp)

## Recocido Simulado

Es un método de búsqueda local, basado en heurísticas, desarrollado por Kirkpatrick, Gellet y Vecchi (1983) y Cerny (1985). Emplea la metáfora del enfriamiento de las moléculas hasta producir una estructura de cristales. Se supone que tenemos un material sometido a temperaturas muy altas y debemos enfriarlo lentamente hasta que se solidifique. Si lo hacemos bruscamente, se endurecerá prematuramente obteniéndose una configuración **sub-óptima**.

# Recocido Simulado

Este método puede inducir a la ocurrencia de un escape a través de un salto a regiones mejores, hasta alcanzar el máximo local.



Fuente: frankfurt-consulting.de

# Recocido Simulado: Generalidades del Algoritmo

- (i)  $T$  indica una temperatura. Se parte de  $T = \infty$  y se debe llegar a  $T = 0$
- (ii)  $x$  indica una solución actual.
- (iii) Hallar un vecino  $y$  de  $x$

Si  $y$  es mejor que  $x$ , entonces se acepta  $y$  como la nueva solución.

Si  $y$  es peor que  $x$  por una cantidad  $\Delta$ , entonces se sustituye  $x$  por  $y$  con probabilidad  $e^{-\Delta/T}$

# Convergencia del Recocido Simulado

- La probabilidad de hacer un movimiento en la dirección equivocada es  $e^{(-\Delta/T)}$ .
- Cuando  $T \rightarrow \infty$ , la prob(direcc. equivocada)  $\rightarrow 1$ .
- Cuando  $T \rightarrow 0$ , la prob(direcc. equivocada)  $\rightarrow 0$ .
- $T$  decrece lentamente de  $\infty$  a  $0$ .
- En teoría, converge hacia la solución óptima.

# Resumen de las Características del Recocido Simulado

- Selección aleatoria de la solución vecina.
- Aceptación probabilística de soluciones que no ofrecen mejoras.
- Se almacena la mejor solución.
- No se mantiene una historia de la búsqueda.
- Se descarta toda la información obtenida durante la búsqueda.

Método de búsqueda desarrollado por Glover y Laguna(1989).

Se basa en un procedimiento determinista, en el cual una “memoria” conduce la búsqueda para explorar nuevas áreas del espacio, evitando regresar a movidas recientemente realizadas (denominadas **tabú**).

# Búsqueda Tabú

La estructura fundamental empleada es la memoria flexible para manejar los atributos de: frecuencia, lo reciente, la calidad e influencia de las soluciones.

Después de evaluar un número de vecindarios, se escoge el mejor.

Acepta movidas de baja calidad.



# Búsqueda Tabú

Emplea la memoria en dos formas:

1. Visitar de nuevo soluciones que ya han sido abordadas.
2. Explorar áreas no visitadas del espacio de soluciones.

# Algoritmo Búsqueda Tabú en pseudolenguaje

Comienzo

$k \leftarrow 1$

Generar una solución inicial  $s$

Mientras no se cumpla la condición de parada

Identificar vecindad  $N(s)$

Identificar Conjunto Tabú  $T(s, k)$

Identificar Conjunto aspirante  $A(s, k)$

Escoger la mejor solución  $s' \in N(s, k) = \{N(s) - T(s, k) + A(s, k)\}$

Memorizar  $s'$ , si mejora la mejor solución previa mejor 'conocida':

$s \leftarrow s'$

$k \leftarrow k + 1$

Fin mientras

Fin

## Comparación entre las técnicas: SA y TS

- Número de vecinos: SA(1 vecino) y TS(n vecinos)
- Acepta las peores movidas?: SA(si, con una probabilidad  $e^{(-\Delta/T)}$  ) y TS (si, el mejor vecino si no es tabú).
- Acepta las mejores movidas?: SA(siempre) y TS(por aspiración).
- Condiciones de parada: SA (baja temperatura, ó si no hay mejora después de cierto número de iteraciones) y TS(número fijo de iteraciones, ó si no hay mejora después de cierto número de iteraciones)

# Resumen SA y TS

- En los métodos tradicionales, de alguna manera siempre se obtiene la misma solución, mientras que en otros, si se parte de soluciones iniciales diferentes, se obtienen resultados diferentes.
- Por otra parte, en los métodos estocásticos, aún con configuraciones iniciales iguales se obtienen soluciones diferentes.
- Todos los métodos tienen en común, que parten de una solución inicial a partir de la cual comienza la exploración.
- En general los métodos o procesan soluciones completas o construyen la solución final a partir de bloques pequeños.
- SA y TS, que son métodos de búsqueda local, pueden detenerse en cualquier momento de la ejecución y dan una solución completa, posiblemente subóptima, esto es debido a que siempre trabajan con la **solución actual mejor** .

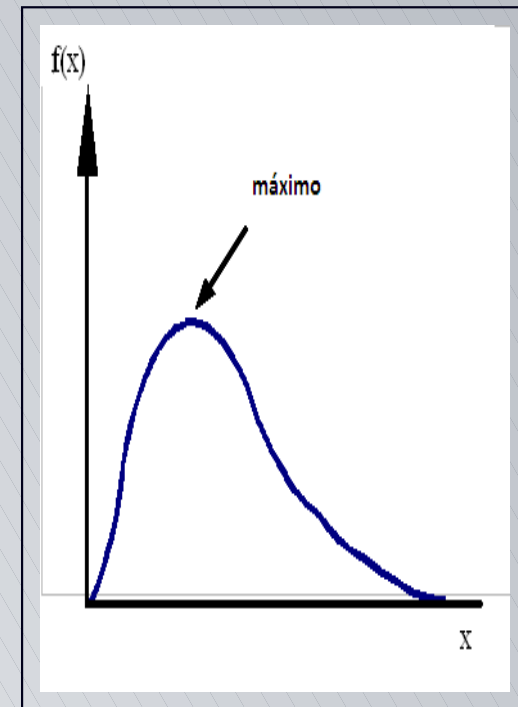
## Método Estocástico de Ascenso de Colinas(SHC,*Hill Climbing*)

- Método de optimización basado en la búsqueda local.
- Es relativamente simple de implementar.
- Tiene buen desempeño en algunos casos.
- Comienza con una solución aleatoria inicial, potencialmente de baja calidad. En cada iteración mejora ligeramente. Cuando detecta que no se producen mejoras, finaliza.
- Puede operar tanto en espacios discretos como en continuos.

# Método de Ascenso de Colinas(*Hill Climbing*)

Este método converge rápidamente a la solución óptima si la función de *fitness* es continua y tiene un solo pico( es unimodal)

En caso de funciones multimodales, probablemente el algoritmo se detiene en el primer pico hallado.



# Algoritmo Ascenso de Colinas Estocástico en pseudo-lenguaje

## Comienzo

$t \leftarrow 0$

Seleccionar un valor  $V_c$  al azar

Evaluar  $V_c$

Repetir

    Seleccionar un punto  $V_n$  de la vecindad de  $V_c$

    Seleccionar  $V_n$  con probabilidad:  $p = \frac{1}{1 + e^{\frac{eval(V_c) - eval(V_n)}{t}}}$

$t \leftarrow t + 1$

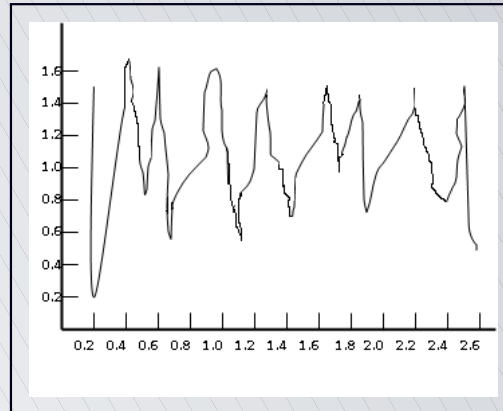
Hasta ( $t = MAX$ )

## Fin

# Resumen: Método de Ascenso de Colinas Estocástico

Opera muy bien en funciones que no poseen muchos óptimos locales.

En caso de funciones ruidosas, es decir aquellas que poseen muchos picos pequeños, definitivamente no es conveniente usarlo.

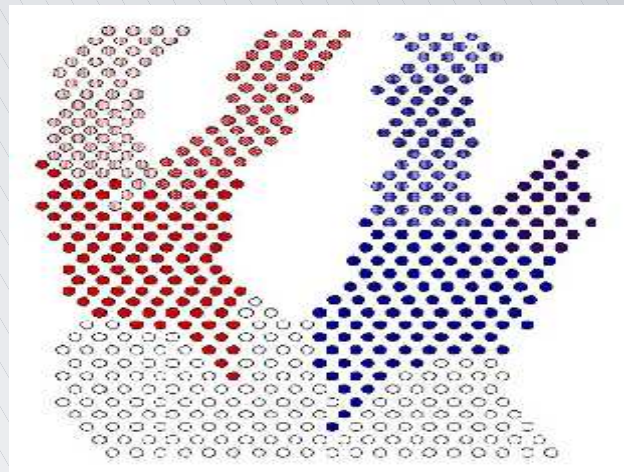


Función con ruido y discontinuidades



# Hacia los métodos de búsqueda bio-inspirados

- Si en vez de procesar una solución a un tiempo, lo hacemos con un conjunto de soluciones simultáneamente, estamos hablando de una **población de soluciones**.
- Se establece una competencia entre las soluciones en la población. Entonces se simula un proceso evolucionista de competencia y selección que permite a las mejores soluciones hallar “espacio” para las nuevas generaciones.
- Es posible usar la variación aleatoria para explorar nuevas soluciones, de manera semejante a la evolución natural.



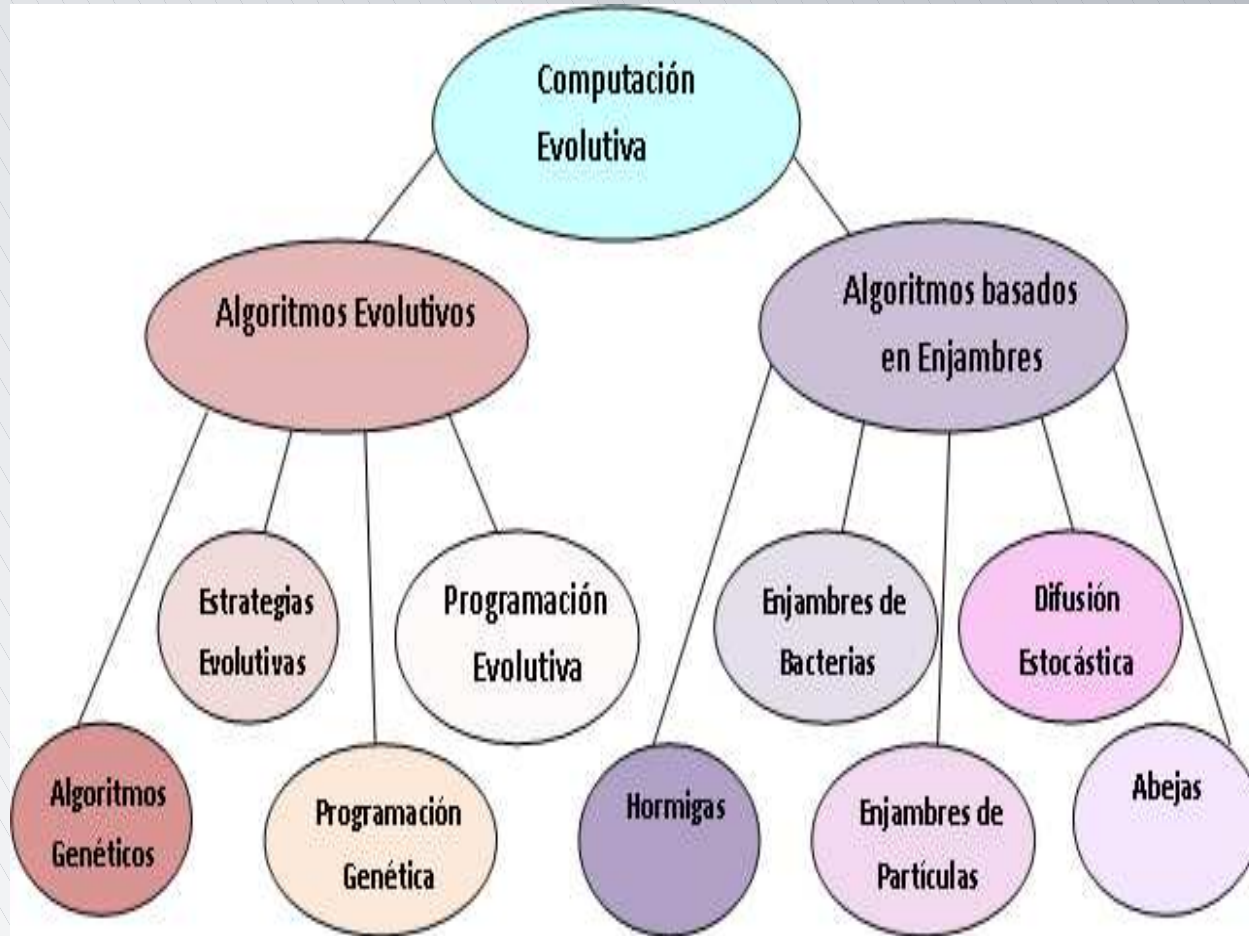
Poblaciones. Fuente: [serendip.brynmawr.edu](http://serendip.brynmawr.edu)

# Elementos de los Métodos Basados en Poblaciones

- El principio de la Selección Natural nos dice que los individuos mejor adaptados son aquellos que sobreviven.
- Toda población de individuos habita en un determinado entorno con recursos limitados.
- Los algoritmos bioinspirados simulan el comportamiento de sistemas naturales y se basan en métodos heurísticos no deterministas de búsqueda de la solución, aprendizaje y comportamiento.
- En términos de espacios de búsqueda, funciones de evaluación y soluciones potenciales, se parte de una población inicial aleatoria en el espacio  $\mathcal{S}$ .
- Las soluciones, en algunos casos, para relacionarlas con la Genética, se tratan como cromosomas o vectores en el contexto computacional.
- Las funciones de *fitness* o evaluación, determinan el mérito de la solución.

La Computación Evolutiva comprende el conjunto de técnicas que permiten resolver una variedad de problemas de optimización. Estas técnicas están basadas en la emulación de la forma de vida y en los mecanismos de organización de seres vivientes para adaptarse al ambiente y evolucionar.

# Taxonomía de Computación Evolutiva



Fuente: Muñoz M. A. *et al*