Máster Universitario en Inteligencia Artificial

MÉTODOS DE SIMULACIÓN

Capítulo 5. Simulación y optimización. Metaheurísticas

POLITÉCNICA
"Ingeniamos el futuro"
CAMPUS
DE EXCELENCIA
INTERNACIONAL

ÍNDICE

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

1. Optimización global y local

Supongamos que queremos resolver el siguiente problema de optimización:

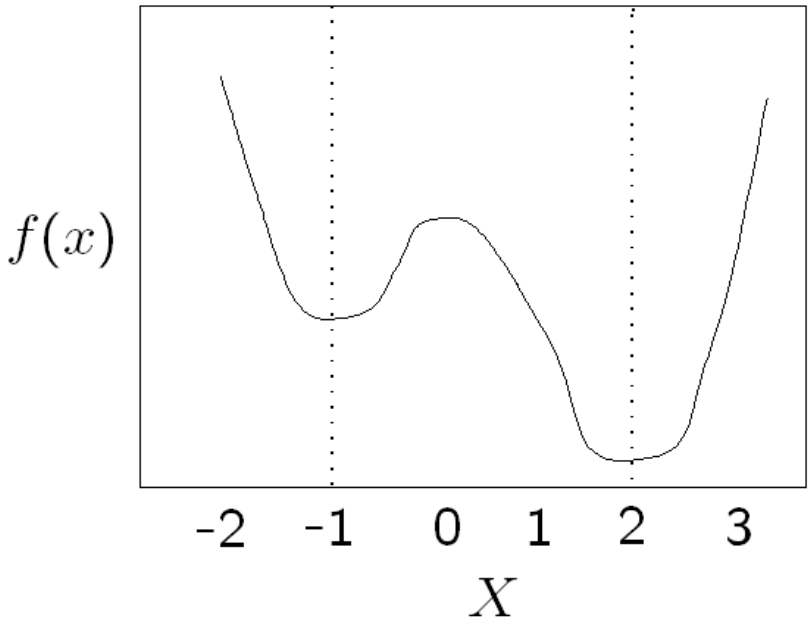
$$\begin{aligned} \min \quad & f(x) \\ \text{s. a} \quad & \\ & x \in X \end{aligned}$$

donde X es el *conjunto factible* (definido por un conjunto de restricciones) y $f: X \rightarrow R$ la *función objetivo*.

$x^* \in X$ es *mínimo global* del problema si $f(x^*) \leq f(x)$, para todo $x \in X$.

$x^o \in X$ es *mínimo local* del problema si $f(x^o) \leq f(x)$, para todo $x \in X$: $0 < \|x - x^o\| \leq \delta$ para cierto $\delta > 0$.

Ejemplo. Sea la función f: $R \rightarrow R$ representada gráficamente en la siguiente figura, tomando como región factible el intervalo [-2, 3]



Dicha función tiene un mínimo local en el punto x = -1, mientras que el mínimo global (dentro de la región factible) se encuentra en el punto x = 2.

Es importante identificar/alcanzar el mínimo global.

Heurísticas más populares → Basadas en la mejora de la *técnica de Búsqueda Local* (*BL*). Procedimiento iterativo en el que partiendo de una solución inicial se va mejorando progresivamente mediante la aplicación de una serie de modificaciones locales o movimientos.

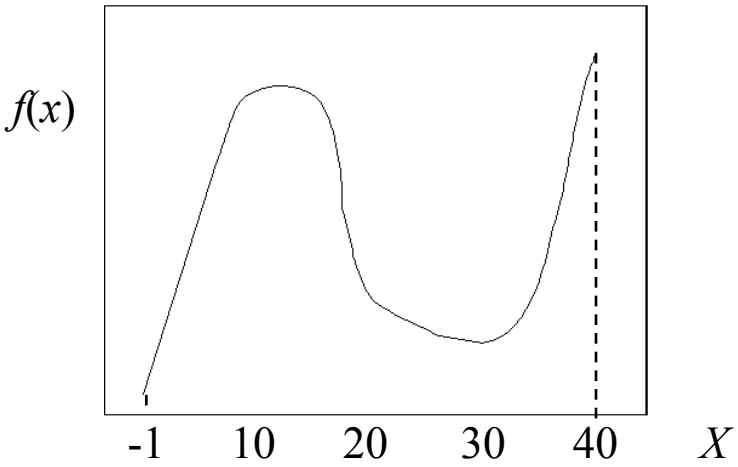
Sean X el conjunto de soluciones factibles, x_i la solución de la iteración i-ésima, $E[x_i]$ un entorno de x_i y x_i^* la mejor solución en $E[x_i]$. El esquema es:

Seleccionar una solución inicial x_1 en X
Hacer $x^* = x_1$, $f^* = f(x_1)$, $i = 1$
Mientras $x_i \neq x_i^*$
  $x_{i+1} = x_i^*$
  $i = i + 1$
  Encontrar x_i^*: $f(x_i^*) < f(x)$, para todo x en $E[x_i]$

Posibilidad → entornos centrados en el punto y con radio r, para la distancia euclídea, es decir, $E[x_n] = [x_n - r, x_n + r]$.
Obviamente, intersecaremos los entornos con la región factible cuando sea necesario.

Ejemplo (*problema continuo*). Supongamos que tenemos el problema de programación no lineal

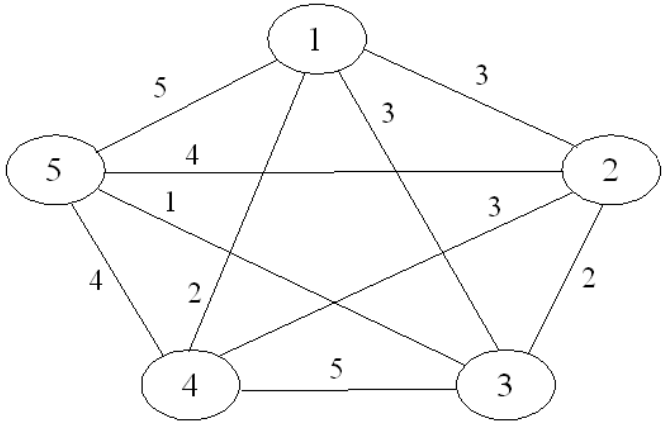
$$\min f(x) = x^3 - 60x^2 + 900x + 100$$
s.a
$$-1 \leq x \leq 40$$



Sean entornos de radio 1 y comenzamos por $x_1 = 20$. Aplicando *BL*, obtenemos $x_2 = 21$, $x_3 = 22$, $x_4 = 23$,..., $x_{11} = 30$. En este punto, para todo $x \in E[x_{11}]$, $f(x) \geq f(x_{11})$, luego $x_{11}^* = x_{11}$ y el algoritmo para.
Obtenemos, por lo tanto, un mínimo local en $x^*=30$, con valor $f^*=100$.

Ejemplo (*problema discreto*). Problema del viajante o del cartero chino (Nemhauser y Wolsey, 1988).

Un viajante, partiendo de una ciudad, debe visitar un conjunto de ciudades volviendo a la ciudad de la que partió, pasando por cada ciudad una única vez y con coste o distancia mínima.



Sea $\mathbf{x} = \{x_1, x_2, x_3, x_4, x_5\}$ una solución factible genérica. *Entorno asociado a* $\boldsymbol{x}$ se define como todos los caminos que se pueden obtener a partir de **x** mediante una trasposición de 2 ciudades.

Sea $\mathbf{x}_1$={1,2,3,4,5}, con $f(\mathbf{x}_1)$=19. La siguiente tabla recoge todas las soluciones factibles pertenecientes al entorno de $\mathbf{x}_1$.

x	$f(x)$	x	$f(x)$
{2,1,3,4,5}	19	{3,2,1,4,5}	12
{4,2,3,1,5}	17	{5,2,3,4,1}	18
{1,3,2,4,5}	17	{1,4,3,2,5}	18
{1,5,3,4,2}	17	{1,2,4,3,5}	17
{1,2,5,4,3}	19	{1,2,3,5,4}	12

La mejor solución en el entorno de $\mathbf{x}_1$ es $\mathbf{x}_2$ = {1,2,3,5,4} (podríamos haber considerado también la solución {3,2,1,4,5}, con el mismo valor en *f*).
Las soluciones factibles en el entorno de $\mathbf{x}_2$ son:

x	$f(x)$	x	$f(x)$
{1,2,3,4,5}	19	{2,1,3,5,4}	14
{3,2,1,5,4}	19	{5,2,3,1,4}	15
{4,2,3,5,1}	13	{1,3,2,5,4}	15
{1,5,3,2,4}	13	{1,4,3,5,2}	15
{1,2,5,3,4}	15	{1,2,4,5,3}	14

Se cumple la condición de parada y $\mathbf{x}_2$ es el mínimo (se trata en realidad del mínimo global).

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

2. Métodos clásicos

Para los problemas de optimización a los que nos enfrentamos no existen condiciones generales de optimización global a diferencia de lo que ocurre con la optimalidad local, como las *condiciones de Karush-Kuhn-Tucker*, lo que dificulta el problema.

Por ello, debemos optar entre:

- *métodos determinísticos* que funcionan sólo bajo condiciones restrictivas (éstos requieren la simplificación del sistema real bajo estudio con el fin de que cumpla condiciones que fundamentan la teoría del modelo en uso, lo que en sistemas complejos podría llevarnos a resolver un sistema muy lejano del real bajo estudio) o

- *modelos estocásticos*, que incorporan algún elemento probabilístico.

Para estos últimos suele utilizarse el término *convergencia probabilística* o *garantía*, que aseguran que bajo ciertas condiciones se obtiene el óptima global con probabilidad alta.

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

Búsqueda aleatoria pura

Consiste en generar aleatoriamente un número grande de soluciones y escoger la mejor.

Sean X el conjunto de soluciones factibles, x_i la solución de la iteración i-ésima, f^* el mejor valor obtenido de f y x^* el valor asociado a f^*. El esquema es:

$f^* = \infty$
Para $i = 1$ hasta N
  Generar $x_i \sim U(X)$
  Si $f(x_i) < f^*$, hacer $f^* = f(x_i)$, $x^* = x_i$

Implementación sencilla si se dispone de un método para muestrear uniformemente de X. El método no es muy eficiente, pues no utiliza información sobre f.

Es sencilla su paralelización y demostración de convergencia bajo condiciones débiles. Dos posibles mejoras son:

1. Utilizar una distribución no uniforme si se dispone de información sobre la localización del óptimo.

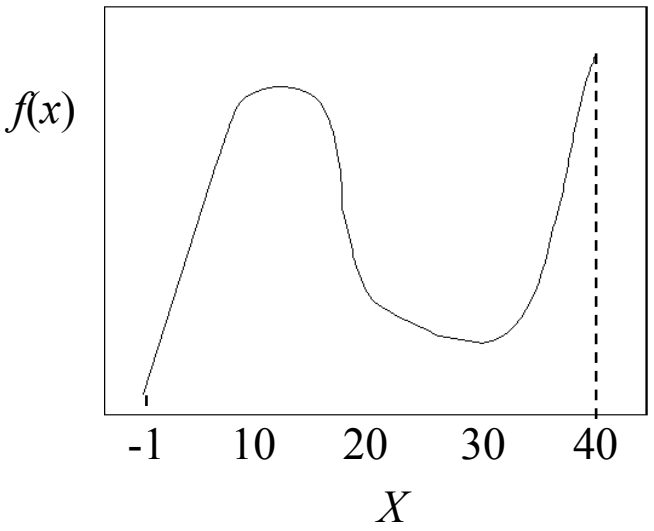
2. Iniciar una optimización local desde el óptimo x^* obtenido.

Ejemplo. Supongamos de nuevo que tenemos el problema de programación no lineal

$$\min f(x) = x^3 - 60x^2 + 900x + 100$$

s.a

$$-1 \leq x \leq 40$$



POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

En la siguiente tabla se muestran 20 valores escogidos de forma aleatoria en el conjunto factible y sus valores asociados para la función a minimizar:

x	12.881	0.637	29.864*	9.607	9.462
$f(x)$	3874.776	649.501	100.551*	4095.309	4091.163
x	8.799	21.957	24.977	23.624	4.581
$f(x)$	4055.043	1520.389	730.017	1060.198	3060.153
x	20.112	31.519	27.170	21.292	38.427
$f(x)$	2066.191	172.777	317.512	1714.280	2828.938
x	8.240	17.931	37.187	9.713	17.750
$f(x)$	4001.702	2711.813	2021.360	4097.511	2763.538

Marcado con un asterisco se muestra el valor que menor valor proporciona para la función objetivo, resultado de la aplicación de este algoritmo, $x^* = 29.86$ y $f^* = 100.55$.

Si aplicamos búsqueda local desde x^* obtenemos el mínimo local $x^* = 30$.

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

Métodos de multicomienzo

La versión básica del método de multicomienzo genera N puntos desde los que se comienza una optimización con un método local proponiendo como solución la mejor así obtenida.

Sean X el conjunto de soluciones factibles, x_i la solución de la iteración i-ésima, f^* el mejor valor obtenido de f y x^* el valor asociado a f^*. El esquema es:

$f^* = \infty$
Desde $i = 1$ hasta N
    Generar $x_i \sim D(X)$
    Aplicar el algoritmo local desde x_i con óptimo x_i^*
    Si $f(x_i^*) < f^*$, hacer $f^* = f(x_i^*)$, $x^* = x_i^*$

D es una distribución que recoge, cuando sea posible, la información disponible sobre el óptimo de f.

Continuación del ejemplo. Generamos 10 valores de forma uniforme del intervalo [-1, 40] y aplicamos el *método de máximo descenso* a partir de cada uno de ellos, obteniendo los resultados que se muestran en la siguiente tabla:

x	33.62	31.16	18.06	28.12	36.2	4.13	5.47	35.05	30.25	27.6
x^*	30	30	30	30	30	-1	-1	30	30	30
f^*	100	100	100	100	100	-861	-861	100	100	100

Por tanto, el mínimo se alcanza en -1.

Posible *inconveniente del método* es que algunas optimizaciones locales que parten de distintas soluciones iniciales pueden conducir al mismo mínimo local.

Se han propuesto varias ideas para mitigar este problema. Las principales son:

- *Reducción*. Se elimina una fracción de las peores soluciones iniciales.

- *Concentración*. Se agrupan las soluciones por proximidad, mediante algún método de *análisis de conglomerados* (Romesburg, 1984), de forma que soluciones.

que pertenezcan a un mismo conglomerado se asimilan a soluciones que conducen a un mismo óptimo.

- *Calentamiento*. Se deja correr la optimización local algunas iteraciones y se aplica, después, alguno de los otros procedimientos.

La demostración de convergencia del método es sencilla.

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

Métodos de direcciones aleatorias

Sean X el conjunto de soluciones factibles, x_i la solución de la iteración i-ésima, f^* el mejor valor obtenido de f y x^* el valor asociado a f^*. La mayoría de los métodos de direcciones aleatorias se adaptan al esquema siguiente:

Seleccionar x_1 en X, $f^* = f(x_1)$, $x^* = x_1$, $i = 1$
Hasta que se satisfaga el criterio de parada
  Generar $\varepsilon_i \sim F_i$
  Hacer $x_{i+1} = D(x_i, \varepsilon_i)$, $i = i + 1$
  Si $f(x_{i+1}) < f^*$, hacer $f^* = f(x_{i+1})$, $x^* = x_{i+1}$

Distintas elecciones de F_i y D conducen a distintos algoritmos. Por ejemplo, si F_i = U(X) y $D(x, \varepsilon) = \text{argmin}\{f(x), f(\varepsilon)\}$, recuperamos el método de búsqueda aleatoria pura.

En muchos casos, se suele escoger F_i como distribución uniforme centrada en x_i y con cierto radio, o normal con media x_i y matriz de varianzas-covarianzas fija.

El nombre del método proviene de que (x, ε) definen una línea aleatoria y, en muchos casos, D se escoge mediante búsqueda lineal en la dirección $(\varepsilon - x)$. En particular, en algunas versiones del algoritmo básico se hace

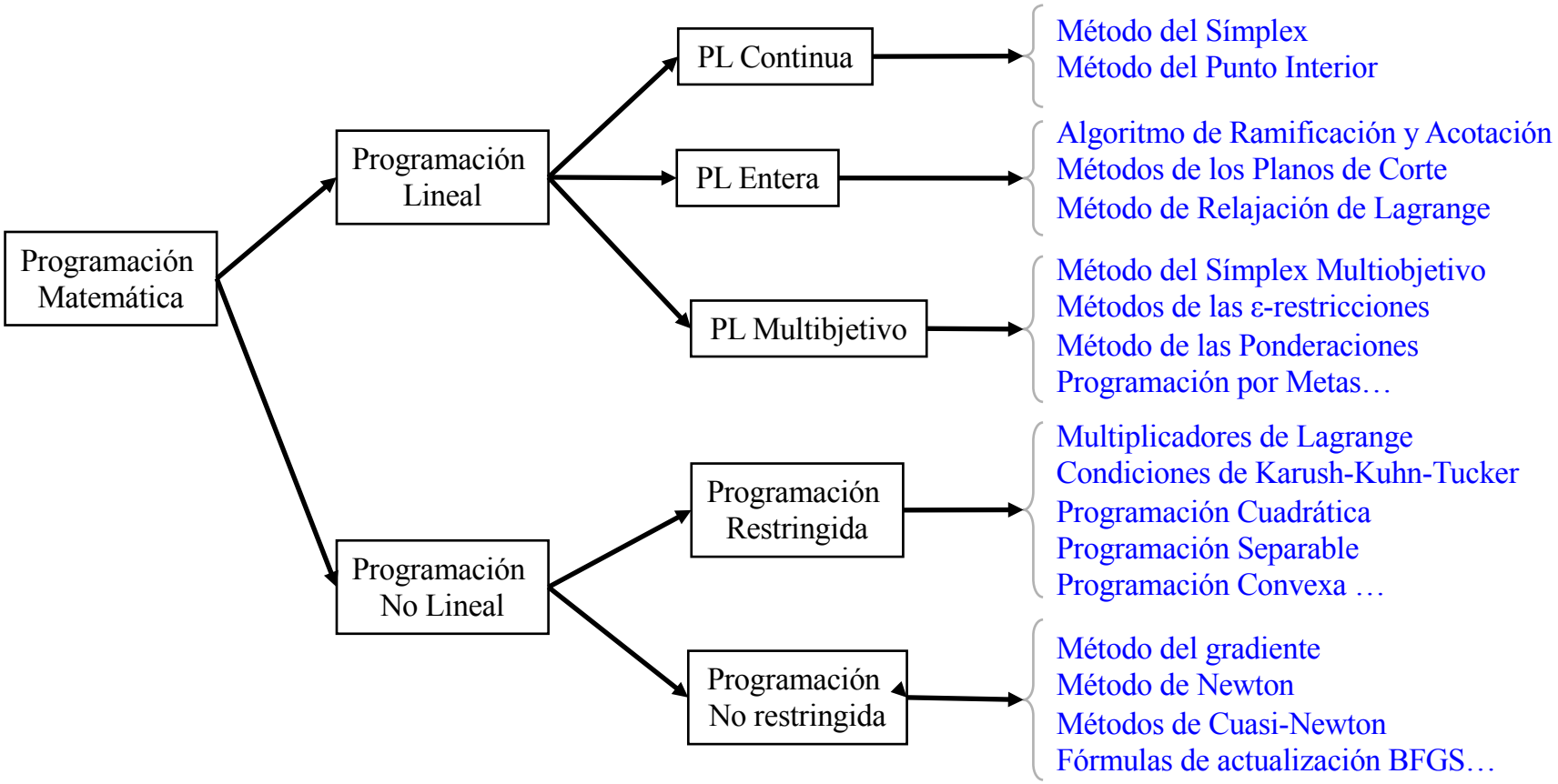
$$D(x, \varepsilon) = x + \alpha^* (\varepsilon - x)$$
$$\alpha^* = \text{argmin}_\alpha f(x + \alpha(\varepsilon - x)),$$

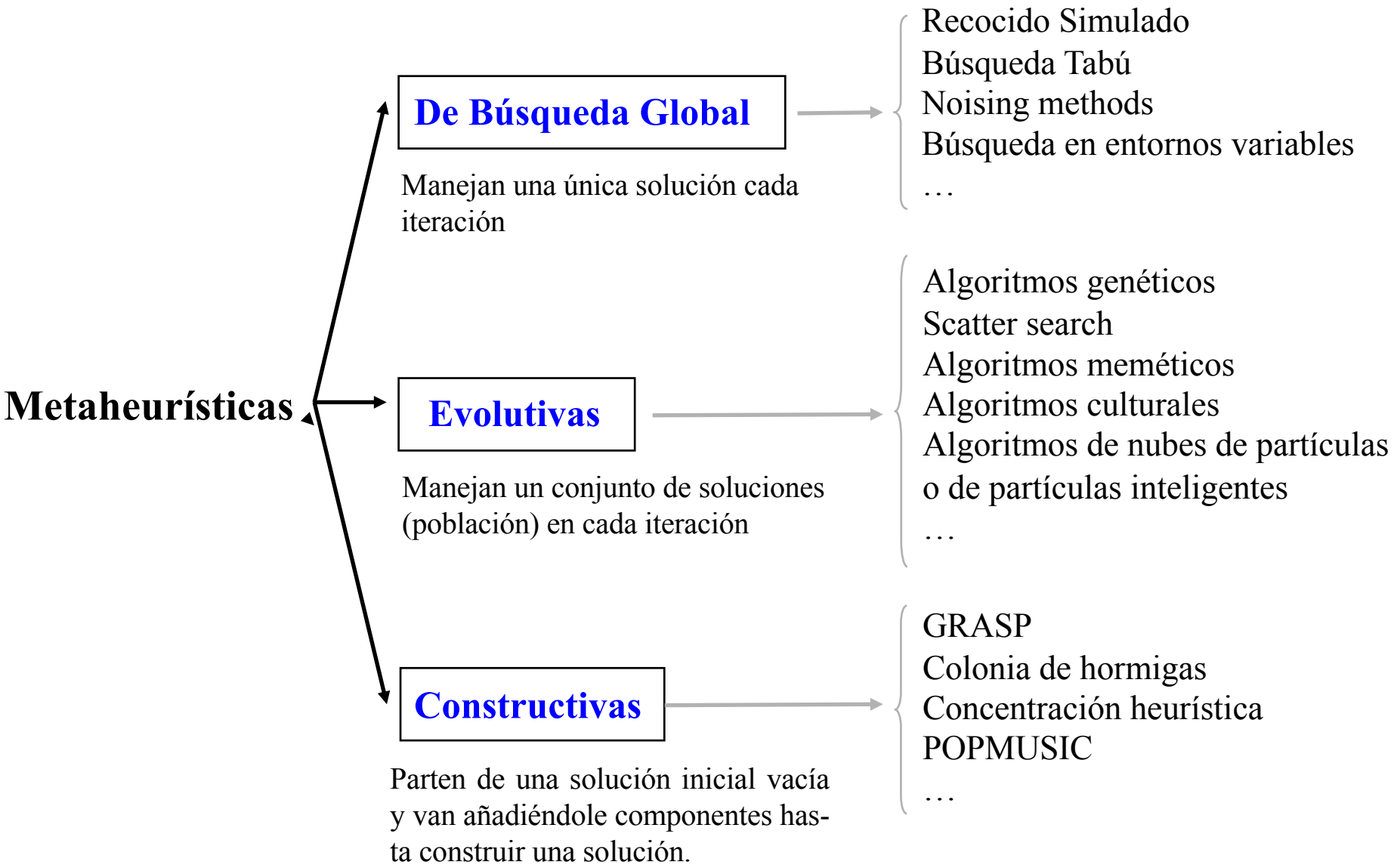
habitualmente con $\alpha \in [0,1]$ ó $\alpha \geq 0$.

La *convergencia* del algoritmo es de demostración sencilla. Sin embargo, resulta complicado proporcionar reglas de parada con garantía de convergencia.

3. Métodos modernos

Existen problemas de optimización tan complejos que no es posible su tratamiento analítico o mediante métodos numéricos. Éstos requieren que se cumplan condiciones que fundamentan la teoría del modelo en uso.




Metaheurísticas
De Búsqueda Global
Manejan una única solución cada iteración
Recocido Simulado
Búsqueda Tabú
Noising methods
Búsqueda en entornos variables
…
Evolutivas
Manejan un conjunto de soluciones (población) en cada iteración
Algoritmos genéticos
Scatter search
Algoritmos meméticos
Algoritmos culturales
Algoritmos de nubes de partículas o de partículas inteligentes
…
Constructivas
Parten de una solución inicial vacía y van añadiéndole componentes hasta construir una solución.
GRASP
Colonia de hormigas
Concentración heurística
POPMUSIC
…

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

Recocido/enfriamiento simulado

Se introduce para superar el inconveniente del método de máximo descenso de quedar atrapado en óptimos locales.

Se debe a Kirpatrick *et al*. (1983) y Cerny (1985), en *Science* y *Journal of Optimization Theory and Application*, respectivamente. En ambos artículos se aplica al problema del viajante. Se ha estudiado bastante el problema de convergencia.

La idea del RS surge de la metalurgia y la termodinámica: mediante un enfriamiento lento se consigue un sólido de entropía mínima.

Al principio se aceptan todos los movimientos, lo que permite explorar todo el espacio de soluciones.

Posteriormente, la temperatura decrece de forma gradual, lo que significa que se hace más selectiva la aceptación de una nueva solución.

Al final, sólo se aceptan movimientos que mejoran la solución actual.

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

El método no busca la mejor solución en un entorno $E(x_i)$ de la solución actual x_i, sino que escoge al azar una solución $y \in E(x_i)$, y

- Si $f(y) \leq f(x_i)$, y se convierte en la actual solución.
- En caso contrario, se selecciona alguna de las dos alternativas siguientes de acuerdo con alguna probabilidad (o ley probabilística)
  - y se convierte en la solución actual con probabilidad $p(i)$
  - x_i permanece como solución actual con prob. complementaria 1- $p(i)$

Típicamente, $p(i)$:

1. decrece con el tiempo (i) al disminuir también la temperatura, y
2. decrece con el deterioro de f ($f(y) - f(x_i)$)

Algoritmo

Inicialización. Escoger una solución inicial $x_1 \in X$. Hacer $x^* = x_1$, $f^* = f(x_1)$, $i = 1$ y escoger $T_1 > 0$.

Pasos $i = 1,2,...$; x_i denota la solución actual

Generar aleatoriamente $y \in E(x_i)$

Si $f(y) < f(x_i)$, hacer $x_{i+1} = y$, *si* $f(y) < f^*$, hacer $x^* = y$, $f^* = f(y)$.

En caso contrario ($f(y) \geq f$), generar $u \sim U(0,1)$.

Si $p(i) = e^{-\left(\frac{f(y)-f(x_i)}{T_i}\right)} > u$, hacer $x_{i+1} = y$

En caso contrario ($p(i) \leq u$), hacer $x_{i+1} = x_i$

Actualizar T_i

Hasta satisfacer el criterio de parada

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

Selección del valor inicial de la temperatura

Si se escoge un valor próximo a cero, no posibilita empeoramiento. Si el valor es muy alto, se ralentiza.

Sugerencia: Tomar T_1 tal que la probabilidad inicial de aceptación de soluciones peores sea alta, por ejemplo, 0.9.

Aceptación de nuevas soluciones

Por analogía con la termodinámica (*distribución de Boltzman*) se suele escoger la probabilidad

$$p(i) = e^{-\left(\frac{f(y)-f(x_i)}{T_i}\right)}$$

donde T_i es la temperatura. Otras expresiones de $p(i)$ en Serafini (1992).

Si T_i es grande, entonces $p(i)$ es grande lo que significa que se aceptan más transiciones a valores peores. Si T_i se reduce, entonces $p(i)$ disminuye y por tanto el número de transiciones.

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

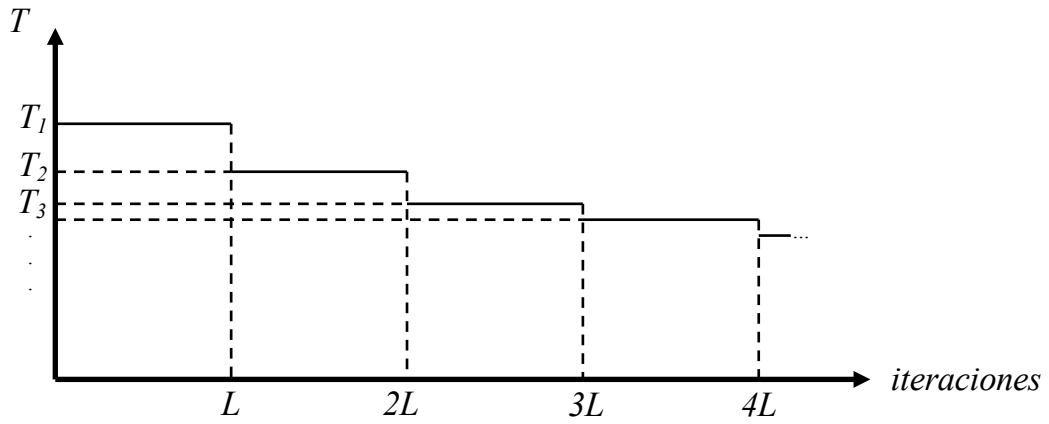
Actualización de la temperatura

Si el decrecimiento es muy lento, la ejecución es lenta.

Si el decrecimiento es muy rápido, se aumenta la probabilidad de quedar atrapado en un mínimo local.

El *esquema habitual* es mantener la temperatura constante durante L iteraciones. Después decrece multiplicándose por $(0 < \alpha < 1)$, de forma que, después de hL pasos la temperatura es $T_{hL} = \alpha^h T_1$.

Un valor típico de α es 0.95. (véase Hajek, 1988).



POLITÉCNICA
"Ingeniamos el futuro"
CAMPUS
DE EXCELENCIA
INTERNACIONAL

Criterio de parada

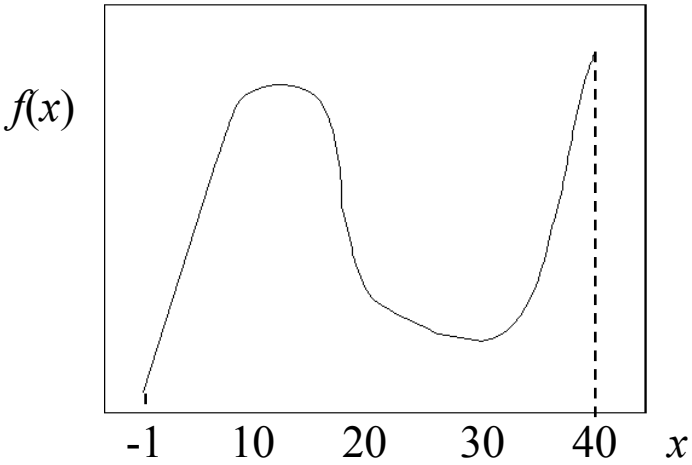
1) Parar si f^* no ha mejorado al menos un $\varepsilon_1\%$, tras k_1 iteraciones de L pasos.

2) Parar si el número de transiciones aceptadas es menor que el $\varepsilon_2\%$ de L tras k_2 iteraciones de L pasos.

Ejemplo (*Recocido Simulado*)

Supongamos que tenemos el problema de programación no lineal

$$\min f(x) = x^3 - 60x^2 + 900x + 100$$
s.a
$$-1 \leq x \leq 40$$



Comenzamos con $x_1 = 20$ y vemos cómo es capaz de salir del mínimo local. Utilizando entornos de radio 2 y probando con varios valores de x_2, decidimos utilizar $T_1 = 4000$. Para el resto de parámetros, consideramos $L = 10$ y $\alpha = 0.9$.

En las siguientes tablas recogemos lo ocurrido en los 20 primeros pasos.

	$T_1 = 4000$					
Iteración	x	$f(x)$	y	$f(y)$	u	$p(i)$
1	20.00	2100.00	20.98	1869.08		
2	20.98		21.60	1623.71		
3	21.60		22.08	1485.92		
4	22.08		20.50	1951.17	.215	.890
5	20.50		21.67	1601.89		
6	21.67		20.55	1935.90	.930	.920
7	21.67		19.78	2163.80	.001	.870
8	19.78		20.48	1956.00		
9	20.48		21.52	1647.70		
10	21.52		21.93	1527.17		

$$p(4) = e^{-\left(\frac{1951.17-1485.92)}{4000}\right)} = 0.89019$$

$T_2 = 3600 = \alpha^1 T_1 = 0.9 \times 4000$						
Iteración	x	$f(x)$	y	$f(y)$	u	$p(i)$
11	*21.93*	*1527.17*	*20.38*	*1987.24*	*.35*	*.88*
12	*20.38*		*20.00*	*2100.00*	*.67*	*.98*
13	*20.00*		*18.66*	*2499.53*	*.43*	*.89*
14	*18.66*		*17.56*	*2817.48*	*.83*	*.92*
15	*17.56*		*18.69*	*2488.92*		
16	*18.69*		*19.18*	*2346.74*		
17	*19.18*		*19.55*	*2235.02*		
18	*19.55*		*20.85*	*1846.00*		
19	*20.85*		*20.34*	*1996.76*	*.59*	*.96*
20	*20.34*		*19.48*	*2256.46*	*.97*	*.93*

Continuamos y tomamos como regla de parada que realice 500 iteraciones más. El resultado obtenido para el mínimo es -860.40, alcanzándose en el punto x^* = -0.9994.

Recordemos que el mínimo global es -1 con valor -861.

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

Algoritmos genéticos

Se introduce para superar el inconveniente del método de máximo descenso de quedar atrapado en óptimos locales.

- Holland (1975), *Adaptation in Natural and Artifical Systems*, The University of Michigan Press.
- Goldberg (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley.

Los algoritmos genéticos son, de hecho, variantes del *método de multicomienzo*.

Por medio de operaciones que derivan de los *principios de la evolución natural*, se identifican soluciones prometedoras, a partir de las que, eventualmente, se puede proceder a realizar una optimización local.

Supongamos que disponemos de una codificación del conjunto factible en un alfabeto finito. Obtener tal codificación puede no resultar sencilla. En el caso continuo, conlleva una discretización de X.

El algoritmo genético parte de un conjunto (*población*) de N soluciones (*individuos*) que evoluciona, manteniéndose su tamaño en las siguientes iteraciones. Es decir, la generación i+1 se obtiene a partir de la i a través de una serie de operaciones.

Designamos mediante $\{x_1^i, ..., x_N^i\} = X^{(i)}$ la población en la iteración i-ésima. Se consideran, entonces, las siguientes operaciones sobre la población $X^{(i)}$:

- **Selección de buenos individuos**. Se escogen de forma determinística o probabilística, y *con reemplazamiento*, los mejores individuos de la población actual (tomando como referencia la función a minimizar).

  Probabilísticamente → $p(\text{escoger } x_j^i) = \dfrac{1/f(x_j^i)}{\sum_l (1/f(x_l^i))}$

  Determinísticamente → Tomar los individuos con menor valor para $f(x_j^i)$

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

- **Cruce de individuos**. Dos individuos parten e intercambian partes de su codificación. La elección de los individuos a cruzar y la selección del *punto de intercambio* se puede hacer de forma determinística o probabilística.

  Supongamos dos individuos con codificaciones $x_j^i = (x_{j1}^i,..., x_{jn}^i)$ e $x_k^i = (x_{k1}^i,..., x_{kn}^i)$ pertenecientes a la población de la iteración i-ésima. Suponiendo que la posición de cruce (escogida de forma aleatoria o probabilística) es la m-ésima ($m < n$), los individuos resultantes serían:

  $$(x_{j1}^i,..., x_{jm-1}^i , x_{km}^i..., x_{kn}^i) \quad \text{y} \quad (x_{k1}^i,..., x_{km-1}^i , x_{jm}^i,..., x_{jn}^i)$$

- **Mutación de un individuo**. Se cambia el valor del algún/os elementos de la codificación de un individuo. La decisión de mutar o no un individuo, la selección del elemento de la codificación sobre la que se realiza la mutación y el nuevo valor que se le asigna a dicho elemento pueden tener carácter probabilístico.

  La mutación se aplica a los hijos o descendientes obtenidos como consecuencia de la operación de cruce, que a su vez se realiza sobre buenos individuos.

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

- **Selección de malos individuos**. Se escogen de forma determinística o probabilística, y *sin reemplazamiento*, los peores individuos de la población actual (tomando como referencia la función a minimizar.)

Probabilísticamente → $p(\text{escoger } x_j^i) = \dfrac{f(x_j^i)}{\sum_l f(x_l^i)}$

Determinísticamente → Tomar los individuos con mayor valor para $f(x_j^i)$

Estos individuos son sustituidos en la siguiente población por los obtenidos como resultado de las operaciones descritas anteriormente.

POLITÉCNICA "Ingeniamos el futuro"
CAMPUS DE EXCELENCIA INTERNACIONAL

Algoritmo

Seleccionar aleatoriamente la población inicial $X^{(1)} = \{x_1^1, ..., x_N^1\}$

Hacer $f^* = \arg\min_i (f(x_i^1))$, $x^* = \min_i (f(x_i^1))$, $j = 1$

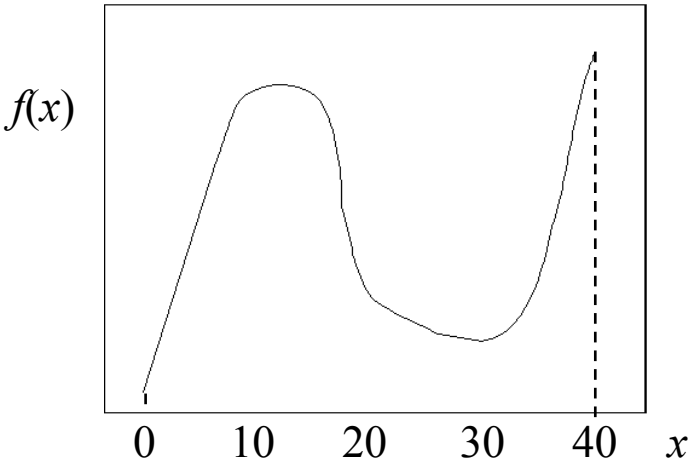
Hasta que se cumpla la condición de parada:

- Seleccionar $2M$ buenos individuos de la población $\{y_1,..., y_{2M}\}$
- Para $k = 1,\ldots,M$ cruzar los pares (y_{2k-1}, y_{2k}) obteniendo los pares $\{z_{2k-1}, z_{2k}\}$
- Para $k = 1,\ldots,2M$ mutar z_k obteniendo w_k. Sea $M_j = \{w_1,\ldots,w_{2M}\}$
- Seleccionar $2M$ malos individuos de la población $V_j = \{v_1,..., v_{2M}\}$
- Hacer $j = j+1$ y $X^{(j)} = (X^{(j-1)} \setminus V_j) \cup M_j$
- Actualizar f^* y x^*.

Ejemplo (Algoritmo genético)

Supongamos que tenemos el problema de programación no lineal

$$\min f(x) = x^3 - 60x^2 + 900x + 100$$
s.a
$$-1 \leq x \leq 40$$



Codificamos las soluciones: Sumamos 1 a cada solución y la multiplicamos por 1000. Estamos considerando el intervalo [0,41] para evitar ambigüedad en la codificación.

Así, cada solución está asociada a cinco valores. Tomamos una población inicial de tamaño $N = 8$:

$$X^{(1)} = \{0.532,\ 8.876,\ 13.751,\ 18.873,\ 25.123,\ 30.752,\ 33.491,\ 39.163\}.$$

$X^{(1)}$

i	Codificación	x_i^1	$f(x_i^1)$	$P(\text{escoger } x_i^1)$
1	0 1 5 3 2	.532	561.969	0.139
2	0 9 8 7 6	8.876	4060.678	0.01716
3	1 4 7 5 1	13.751	3730.676	0.01861
4	1 9 8 7 3	18.873	2436.668	0.02849
5	2 6 1 2 3	25.123	697.553	0.09955
6	3 1 7 5 2	30.752	117.390	0.59157
7	3 4 4 9 1	33.491	508.157	0.13665
8	4 0 1 6 3	39.163	3388.147	0.02049

$$\frac{1/561.969}{1/561.969+\ldots+1/3388.147} \approx 0.139$$

x_6^1 3 1 7 5 2

x_1^1 0 1 5 3 2

x_6^1 3 1 7 5 2

x_6^1 3 1 7 5 2

Selección de 4 ($M = 2$) buenos individuos:

Determinísticamente:

Probabilísticamente:

Formamos dos pares al azar y realizamos las **operaciones de cruce y mutación**:

(1) Para seleccionar la posición de cruce obtenemos dos números aleatorios de una U[0,1]. Suponiendo que se tratan de los valores 0.6166 y 0.5822, en ambos casos el intercambio se intercambia a partir del tercer elemento de la codificación.

Buenos individuos		Cruce	Mutación	w_i	$f(w_i)$
x_6^1	3 1 7 | 5 2	3 1 7 3 2	3 1 7 3 2	30.735	561.969
x_1^1	0 1 5 | 3 2	0 1 5 5 2	0 1 5 5 2	0.552	578.685
x_6^1	3 1 7 | 5 2	3 1 7 5 2	3 1 7 5 2	30.752	117.390
x_6^1	3 1 7 | 5 2	3 1 7 5 2	3 1 7 8 2	30.782	123.497

(2) Obtenemos cuatro valores de la distribución U[0,1] para ver sobre que individuos realizo mutación (0.013, 0.977, 0.256, 0.640). Los valores superiores a 0.5 me indican los individuos sobre los que realizar la mutación.

A continuación, selecciono el elemento de la codificación que voy a mutar para cada individuo (0.360 → posición 2 y 0.788 → posición 4) y también de forma aleatoria elijo el nuevo valor que tomará el elemento de la codificación. (0.1077 → valor 1, 0.819 → valor 8).

Seleccionamos los 4 **peores individuos** de la población:

$X^{(1)}$

i	Codificación	x_i^1	$f(x_i^1)$	$P(\text{escoger } x_i^1)$
1	0 1 5 3 2	.532	561.969	0.036
2	0 9 8 7 6	8.876	4060.678	0.2619
3	1 4 7 5 1	13.751	3730.676	0.2406
4	1 9 8 7 3	18.873	2436.668	0.1572
5	2 6 1 2 3	25.123	697.553	0.045
6	3 1 7 5 2	30.752	117.390	0.0075
7	3 4 4 9 1	33.491	508.157	0.0327
8	4 0 1 6 3	39.163	3388.147	0.2185

$$\frac{561.969}{561.969+\ldots+3388.147} \approx 0.03639$$

$$V_j = \begin{cases} x_2^1 & 0\,9\,8\,7\,6 \\ x_7^1 & 3\,4\,4\,9\,1 \\ x_4^1 & 1\,9\,8\,7\,3 \\ x_8^1 & 4\,0\,1\,6\,3 \end{cases}$$

Determinísticamente:

Probabilísticamente:

Calculamos la nueva población $X^{(j+1)} = (X^{(j)} \setminus V_j) \cup M_j$

$X^{(1)}$

i	Codificación	x_i^1	$f(x_i^1)$
1	0 1 5 3 2	.532	561.969
~~2~~	~~0 9 8 7 6~~	~~8.876~~	~~4060.678~~
3	1 4 7 5 1	13.751	3730.676
~~4~~	~~1 9 8 7 3~~	~~18.873~~	~~2436.668~~
5	2 6 1 2 3	25.123	697.553
6	3 1 7 5 2	30.752	117.390
~~7~~	~~3 4 4 9 1~~	~~33.491~~	~~508.157~~
~~8~~	~~4 0 1 6 3~~	~~39.163~~	~~3388.147~~

Buenos individuos		Cruce	Mutación
x_6^1	3 1 7 | 5 2	3 1 7 3 2	3 1 7 3 2
x_1^1	0 1 5 | 3 2	0 1 5 5 2	0 1 5 5 2
x_6^1	3 1 7 | 5 2	3 1 7 5 2	3 1 7 5 2
x_6^1	3 1 7 | 5 2	3 1 7 5 2	3 1 7 8 2

$X^{(2)}$

i	Codificación	x_i^2	$f(x_i^2)$
1	0 1 5 3 2	0.532	561.969
2	3 1 7 3 2	30.732	135.366
3	1 4 7 5 1	13.751	3730.676
4	0 1 5 5 2	0.552	578.685
5	2 6 1 2 3	25.123	697.553
6	3 1 7 5 2	30.752	117.390
7	3 1 7 5 2	30.752	117.390
8	3 1 7 8 2	30.782	123.497

4. Referencias

Cerny, V. (1985). A Thermodynamical Approach to the Travelling Salesman problem: An Efficient simulated Algorithm, Journal of Optimization Theory and Application **45**, 41-51.

Charon, I. y Hudry, O. (2001a). The Noising methods: A Survey, in C.C. Ribeiro y P. Hansen (eds.), Essays and Surveys in Metaheuristics, 469-492, Kluwer Academic Publishers. (disponible en Aula Virtual)

Charon, I. y Hudry, O. (2001b), The noising methods: A generalization of some metaheuristics, European Journal of Operational Research **135**, 86-101. (disponible en Aula Virtual)

Coello C., Van Veldhuizen D. y Lamont G. (2002). Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers.

Engelbrecht, A.P. (2005), Fundamentals of Computational Swarm Intelligence, John Wiley & Sons.

De Jong, K.A. (2006). Evolutionary computation: a unified approach, MIT Press.

Dorigo, M., Maniezzo, V. y Colorni, A. (1996). *The Ant System: Optimization by a Colony of Cooperating Agents*, IEEE Tranactions on Systems, Management and Cybernetics-Part B **26**, 1-13. (disponible en Aula Virtual)

Dorigo, M. y Stutzle, T. (2003). The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances, in F. Glover and G. Kochenberger (eds.) Handbook of Metaheuristics, 251-285, Kluwer Academic Publishers.

Dorigo, M. y Stützle, T. (2004). Ant Colony Optimization, The MIT Press.

Feo, T.A. y Resende, M.G.C. (1995). Greedy Randomized Adaptive Search Procedures, Journal of Global Optimization **6** (2), 109-133.

Glover, F. (1977). Heuristics for Integer Programming using Surrogate Constraints, Decision Science **8**, 156-166.

Glover F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research **13**, 533-549.

Glover F. (1989). Tabu Search- Part I., ORSA Journal on Computing **1**, 190-206.

Glover F. (1990). Tabu Search- Part II., ORSA Journal on Computing **2**, 4-32.

Glover, F. y Kochenberg, G.A. (1993). Handbook of Metaheuristics. Operations Research Management Science, Kluwer Academic Publishers.

Glover, F. y Laguna, M. (1997). Tabu Search, Kluwer Academic Publishers.

Glover, F. y Melián, B. (2003). Tabu Search, Revista Iberoamericana de Inteligencia Artificial **19**, 29-48. (disponible en Aula Virtual)

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley

Hajek, B. (1988). Cooling schedules for optimal annealing, Mathematical Operations Research **13**, 311-329.

Hansen, P., Mladenovic, N. y Moreno, J.A. (2003). Variable Neighbourhood Search, Revista Iberoamericana de Inteligencia Artificial **19**, 77-92. (disponible en Aula Virtual)

Holland, J.H. (1975). Adaptation in Natural and Artificial Systems, The University of Michigan Press.

Kennedy, J. y Eberhart, R.C. (1995), Particle Swarm Optimization, Proceedings 1995 IEEE International Conference on Neural Networks, 1942-1948, IEEE Press. (disponible en Aula Virtual)

Kennedy, J. y Eberhart, R.C. (2001), Swarm Intelligence, Morgan Kauffmann.

Kirkpatrick, S., Gelatt, C.D. y Vechi, M.P. (1983). Optimization by Simulated Annealing, Science **220**, 671-680.

Larranaga, P. y Lozano, J.A. (eds.) (2001), Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Springer.

Manrique, D. y Ríos, J. (2006). Redes de Neuronas Artificiales y Computación Evolutiva, Facultad de Informática, Universidad Politécnica de Madrid.

Moscato, P. (1999). Memetic Algorithms: A Short Introduction, in D. Corne, M. Dorigo, F. Glover (eds.), New Ideas in Optimization, 219-234, McGraw Hill.

Moscato, P. y Cotta, C. (2003). A Gentle Introduction to Memetic Algorithms, in F. Glover, G. Kochenberger (eds.), Handbook of Metaheuristics, 105-144, Kluwer Academic Publishers.

Moscato, P. y Cotta, C. (2003). Una Introducción a los Algoritmos Meméticos, Revista Iberoamericana de Inteligencia Artificial **19**(2), 131-148. (disponible en Aula Virtual)

Nemhauser, G. y Wolsey, L. (1988). Integer and Combinatorial Optimization, Wiley.

Reynolds, R.G. (1994). An Introduction to Cultural Algorithms, Proceedings of the 3rd Annual Conference on Evolutionary Programming, World Scienfific Publishing, 131–139. (disponible en Aula Virtual)

Resende, M.G.C. y Ribeiro, C.C. (2003). Greedy randomized adaptive search procedures, in F. Glover y T. Kochenberger (eds.), Handbook of Metaheuristics, 219-249, Kluwer. (disponible en Aula Virtual)

Romesburg, H.C. (1984). Cluster Analysis for Researchers, Lifetime Learning Pub.

Rosing K.E. y ReVelle C.S. (1997). Heuristic concentration: two stage solution construction, European Journal of Operational Research **97**, 75-86. (disponible en Aula Virtual)

Serafini, P. (1992). Simulated Annealing for Multiple Objective Optimization Problems, in G.H. Tzeng, H.F. Wang, V.P. Wen and P.L. Yu, (eds.), Proceedings of the Tenth International Conference on MCDM, 87-96, Springer Verlag.

É.D. Taillard y S. Voss (2001). POPMUSIC: Partial Optimization Metaheuristic Under Special Intensification Conditions, in C. Ribeiro y P. Hansen (eds.), Essays and surveys in metaheuristics, 613-629, Kluwer.