

Metaheurísticos para resolver problemas muy complejos



Métodos de Optimización

Los métodos para resolver problemas de Optimización pueden clasificarse en:

- **Exactos:** Garantizan encontrar el óptimo global
 - Desventaja: ineficiencia por la alta dimensionalidad
- **Heurísticos o metaheurísticos:** solo permiten aproximarse a los óptimos globales
 - Ventaja: más eficientes y flexibles

Metaheurísticos

*Los procedimientos Metaheurísticos son una clase de **métodos aproximados** que están diseñados para resolver **problemas difíciles de optimización combinatoria**, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Los Metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de: inteligencia artificial, evolución biológica y mecanismos estadísticos"*

Osman y Kelly (1995)

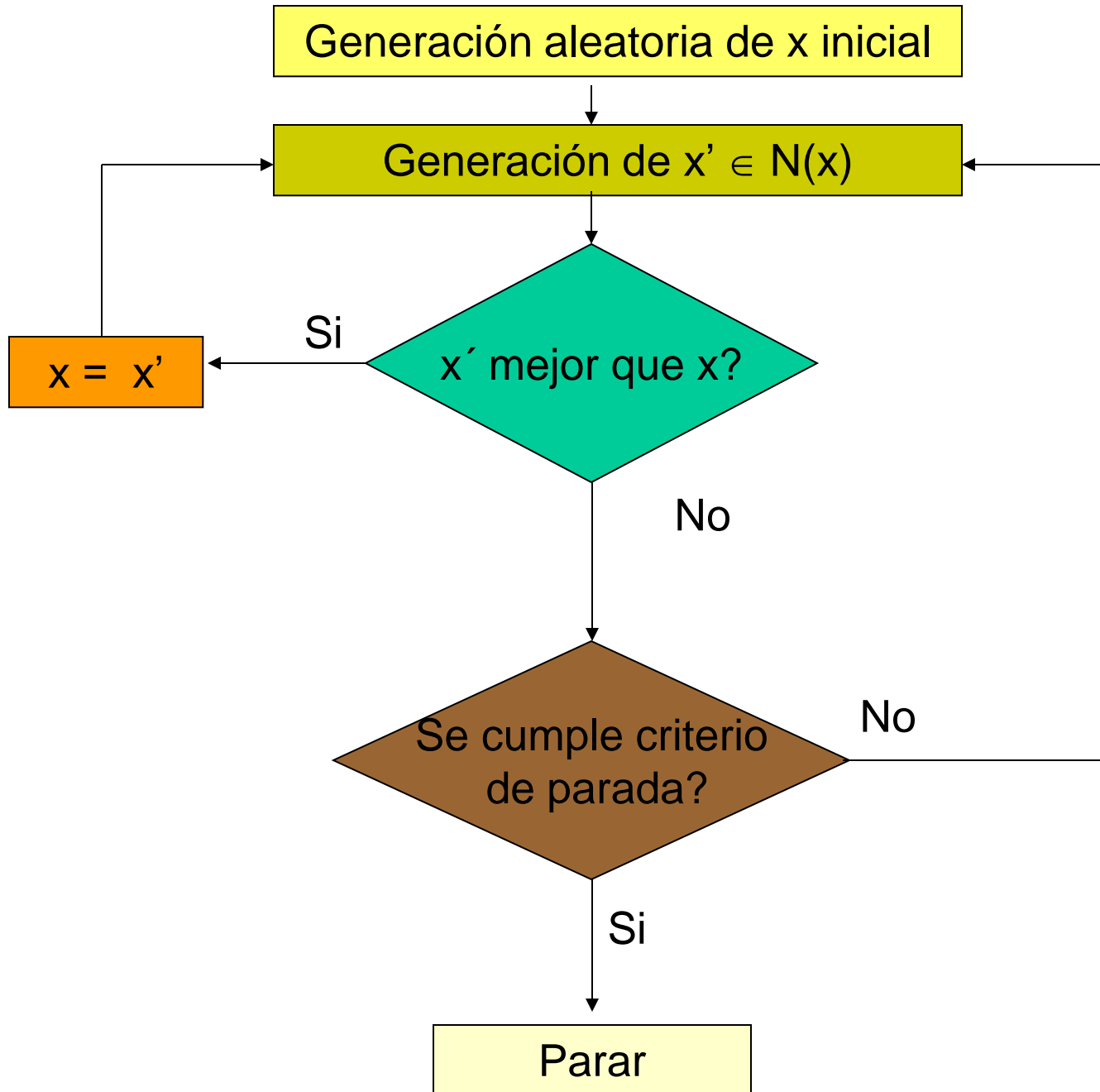
Metaheurísticos

- Son basados en principios generales de Inteligencia Artificial (IA).
- No garantizan la optimalidad de la solución encontrada
- Su propósito es encontrar una solución cercana al óptimo en un tiempo razonable.

Metaheurísticos

En los últimos años ha habido un crecimiento espectacular en el desarrollo de procedimientos metaheurísticos

Han aparecido publicaciones específicas para el estudio y divulgación de dichos procedimientos tales como *Journal of Heuristics*.



Estos tipos de procesos pueden caer en mínimos locales

Opciones:

- Ejecutar el algoritmo a partir de muchos puntos iniciales y salvar los mejores resultados.
- Aceptar la nueva solución aunque no supere a la anterior según algunas reglas
- Usar reglas de transición probabilísticas, no determinísticas

Grados de Complejidad Computacional

- **Problemas tipo P**

El número de iteraciones requeridas para su solución es proporcional al tamaño del problema (no son difíciles de solucionar)

- **Problemas tipo NP**

El número de iteraciones requeridas para su solución no es proporcional al tamaño del problema

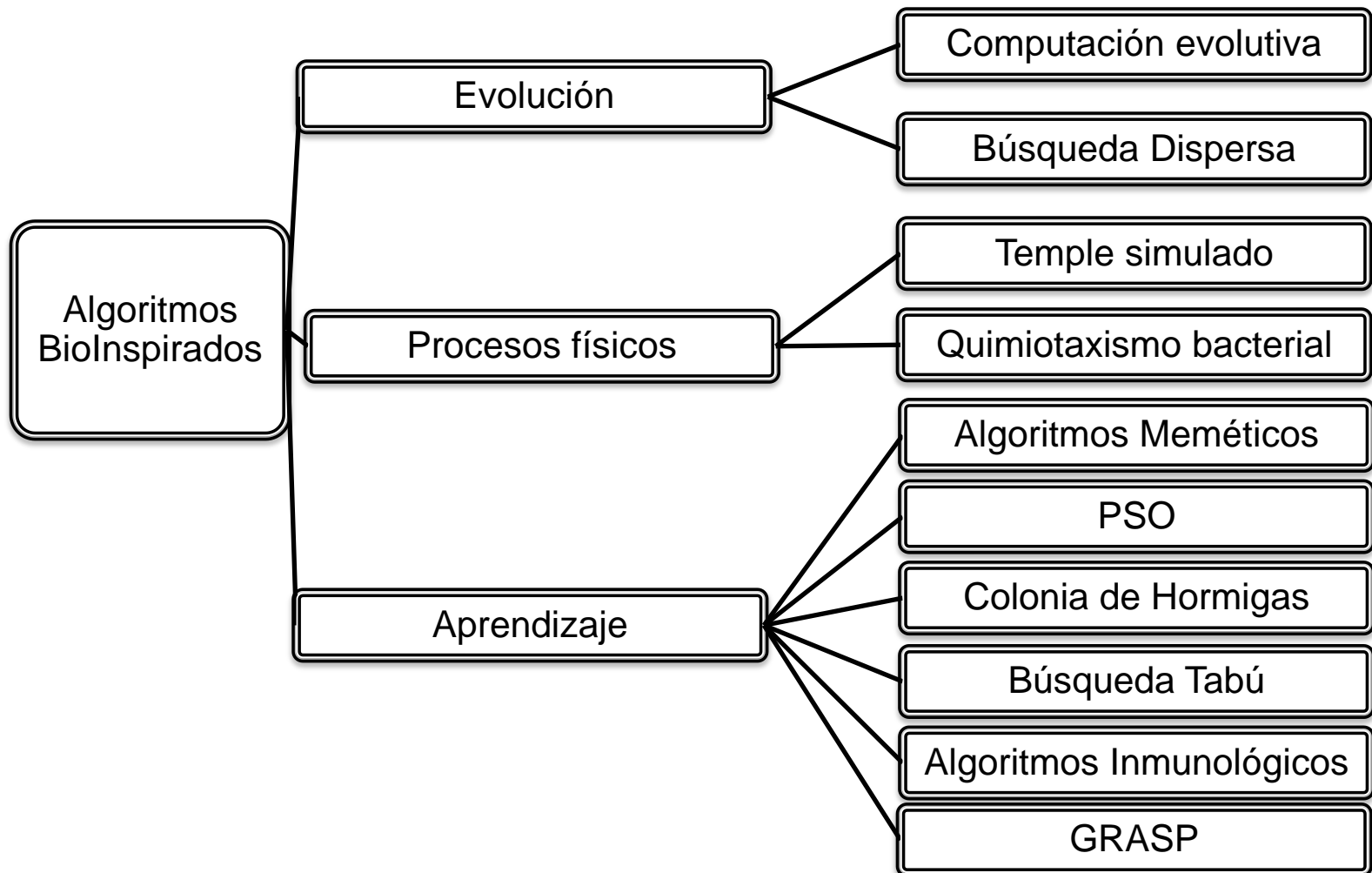
(son muy difíciles de solucionar)

Casos en los que es interesante usar Metaheurísticas

- Cuando la solución requiere de mucho tiempo computacional
- Cuando no se necesita la solución Óptima
- Cuando los datos no son fiables
- Ante limitaciones de tiempo o espacio para almacenamiento de datos
- Como paso intermedio en la aplicación de otro algoritmo

Métodos Meta-heurísticos

Según su Inspiración pueden clasificarse en:



Aplicaciones

- Secuenciación

- *Scheduling*
- Planificación de fuerza laboral
- Horarios escolares

- Diseño

- Redes de transporte
- Planificación del espacio en arquitectura
- Distribución en Planta

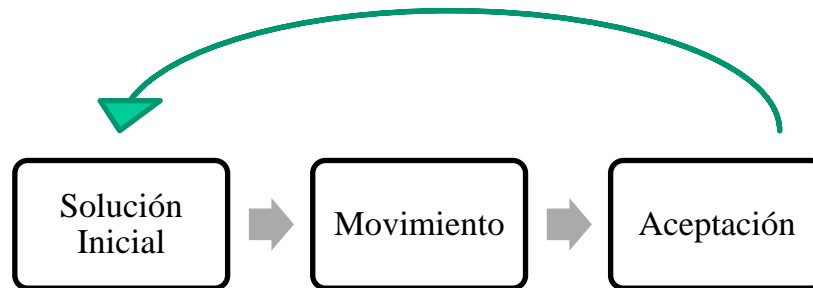
Lógica e Inteligencia Artificial

- Reconocimiento de patrones
- Entrenamiento de Redes neuronales

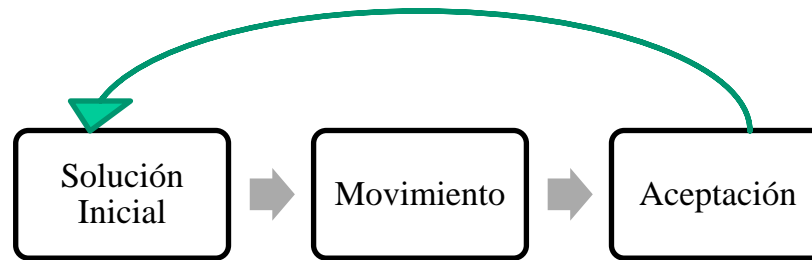
- Tecnología

- Distribución de energía eléctrica
- Diseño estructural de Ingeniería
- Explotación de crudo

Descripción de los métodos(2 de 5)



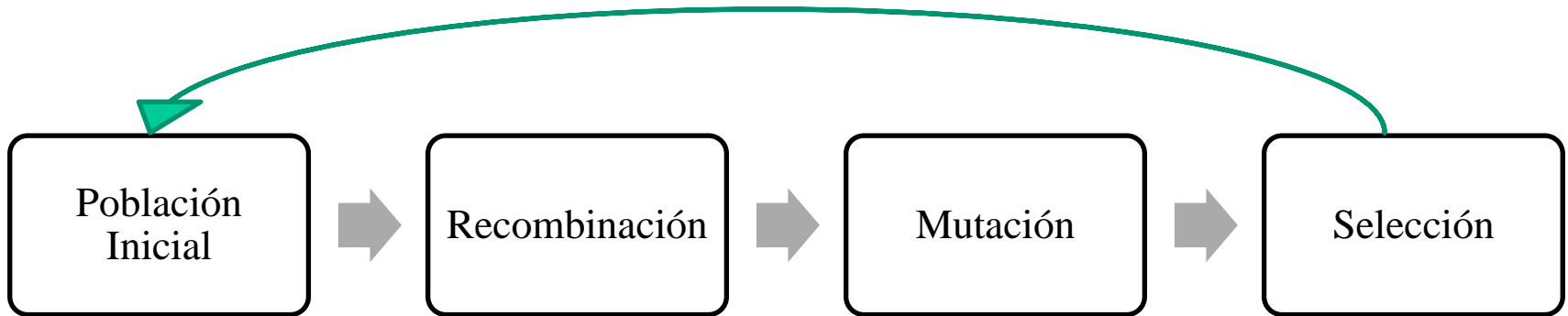
- ***Temple simulado***: propuesto en los 70's por Kirkpatrick, simula el proceso de refinación de grano del acero en el proceso de temple (Funciones: T, E; Parámetros: $T(0), \mu$)



- **Quimiotaxismo Bacterial:** Propuesto en el 2000 por Müller *et al*(2000), simula la forma como las bacterias se orientan según la concentración de alimento del ambiente. (Funciones: $\tau(T)$, $\theta(\tau)$; Parámetros: $T(0)$, $\tau(0), v=1$)[18]



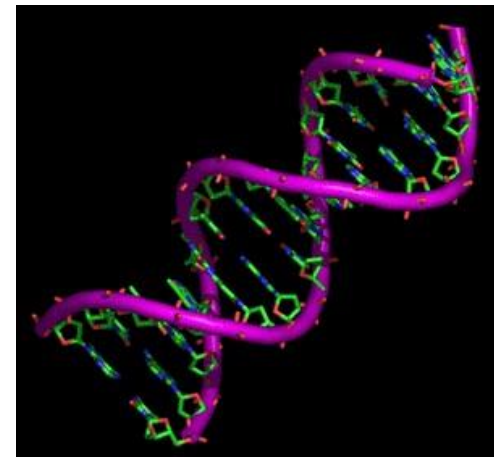
Inspirados en la evolución

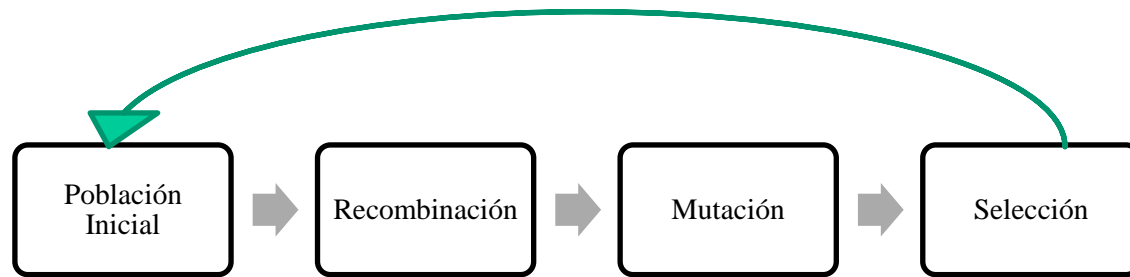


– ***Computación evolutiva:*** se divide en 4 paradigmas

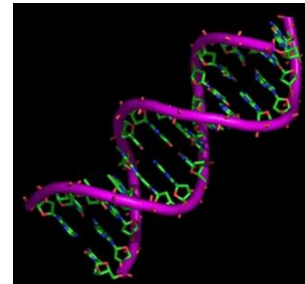
- Estrategias Evolutivas
- Algoritmos genéticos
- Evolución diferencial
- Programación evolutiva

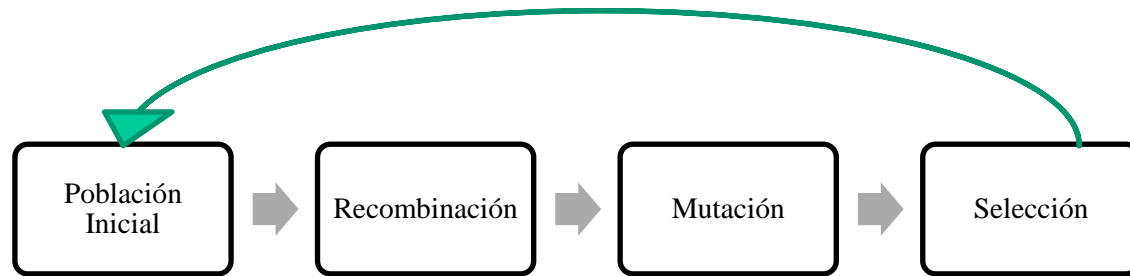
– ***Búsqueda dispersa***



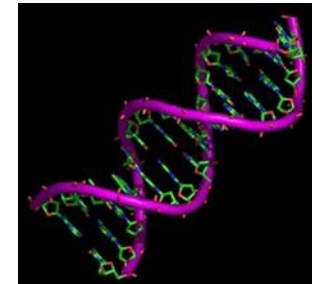


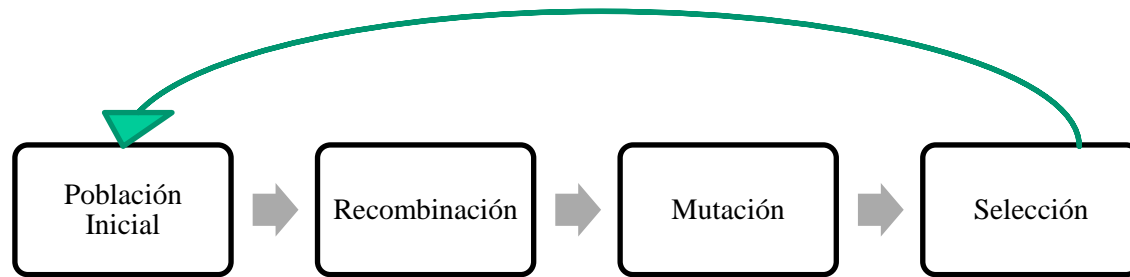
- Estrategias Evolutivas: Propuesto por Rechember y Schewefel en la década de los 60's, inspirada en el proceso de evolución con una adaptación incipiente. (Optimización continua; Funciones: mutación, recombinación; Parámetros: # de padres, # de hijos, factores de recombinación y mutación)[20,22,23]





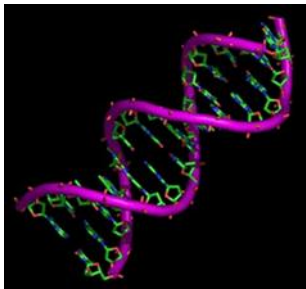
- Algoritmos genéticos: Propuestos inicialmente por John Holland en la década de los 60's, tienen su inspiración en el proceso de selección natural en el mejoramiento de una población basado en la reproducción sexual y la mutación. (Optimización binaria; Funciones: escalamiento función objetivo; Parámetros: # de individuos de la población, factores de recombinación)

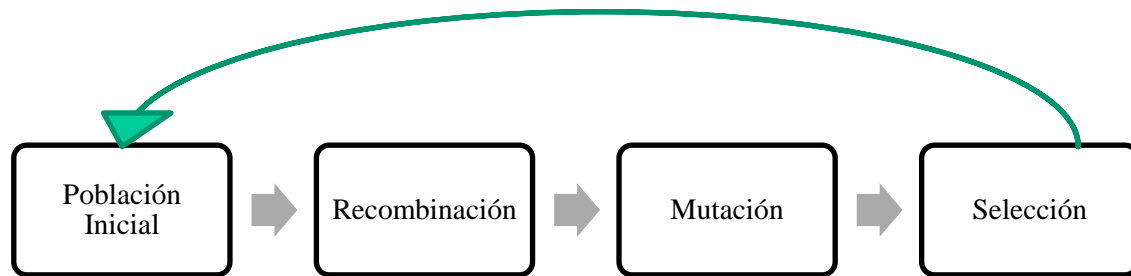
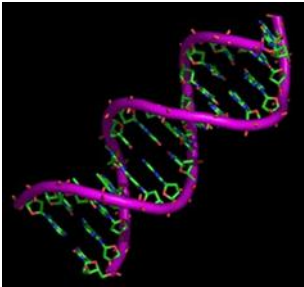




- Evolución diferencial: se propuso por Storn and Prince en 1995.

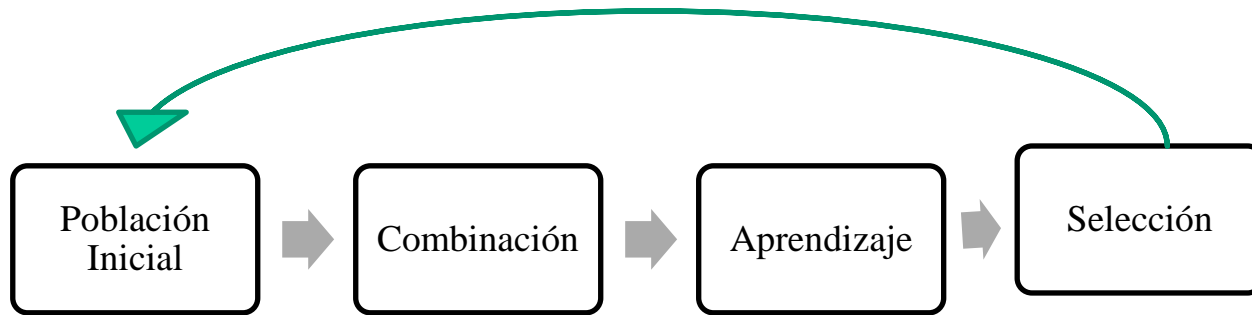
Difiere respecto a sus predecesores en la forma de conseguir la herencia, pues los operadores normales de recombinación o mutación se unen en una vectorización. (Optimización real; Funciones: función de densidad probabilística; Parámetros: # de individuos de la población, factores para la vectorización, bordes superior e inferior de la región de búsqueda)





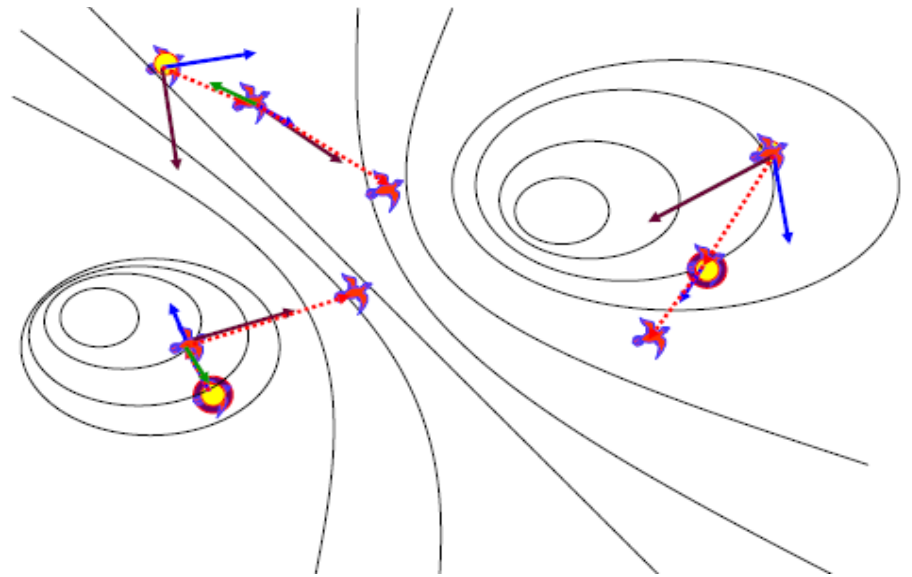
- Programación evolutiva: Propuesto por Fogel en los 60's. Recibe su inspiración del comportamiento inteligente que permite predecir un entorno y responder a él de acuerdo a un objetivo trazado. (Optimización real o discreta; Parámetros: # de individuos de la población, # de individuos Q, factores de mutación y combinación)

Inspirados en el aprendizaje



- PSO
- Colonia de Hormigas
- Búsqueda Tabú
- Algoritmos Inmunológicos
- GRASP

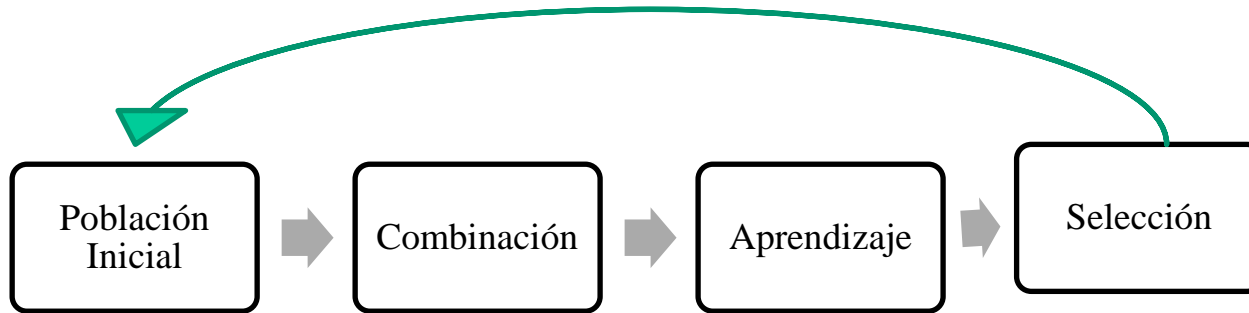
- **PSO:** Propuesto por Eberhart and Kennedy en 1995. Se inspira en el comportamiento gregario presente en la naturaleza, donde se proveen dos tipos de conocimientos, el particular y el colectivo. (Optimización real; Funciones: v y p ; Parámetros: # de individuos de la población, Pesos de los movimientos $v(0)$)





- ***Colonia de Hormigas:*** Propuesto por Dorigo a principios de los 90's. Se inspira en la comunicación indirecta de una colonia de hormigas, emulando el funcionamiento de las feromonas, donde las mejores soluciones son marcadas y recordadas en la memoria colectiva. (Optimización combinatoria; Parámetros: # de individuos de la población, Pesos de los feromonas individuales y globales)[9,31]





- ***Busqueda tabú***: Propuesta por Glover y Laguna en 1986. Inspirada en la memoria, hace una búsqueda inteligente evitando las soluciones prohibidas, trata de buscar exhaustivamente, de manera determinista. (Optimización combinatoria; Parámetros: Tamaño de la lista, argumentos de control de la lista tabú, solución inicial)[10,27]

Tendencias Actuales

- Metaheurísticas Híbridas
- Coevolución
- Robótica evolutiva
- Nuevas metaheurísticas (evolución diferencial, optimización mediante cúmulos de partículas, etc.).

Conclusiones

- La metaheurísticas ha mostrado ser una alternativa viable para resolver problemas del mundo real de alta complejidad.
- Su uso, sin embargo, debiera limitarse a problemas de alta complejidad en los cuales ninguna otra técnica (determinista) funciona.

Referencias

- [1] J. Vesterstrom, R.Thomsen. "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems", Proceedings of congress of evolutionary computation, Portland, 2004, pp. 1980-1987. Vol 2.
- [2] W., Sun, Y.X, Yuan. Optimization Theory and Methods, Nonlinear programming, Springer, New York, 2006.
- [3] S, Rao, Engineering Optimization, Theory and practice. John Wisley & sons, Estados Unidos, 1996.
- [4] J.S.R., Jang, C.T. Sun, E., Mizutan, Neuro-fuzzy and soft computing. Prentice Hall, Upper Saddle River, 1996.
- [5] D.A. Pierre, Optimization theory with applications. Dover, New York, 1986.
- [6] S., Nash, A., Sofer, Linear and Nonlinear Programming. McGraw hill, New York, 1996
- [7] T. Bäck, D.B. Fogel, and Z. Michalewicz, Evolutionary Computation 1, IOP Press, Bristol, 1997.
- [8] P. Cortés, J. García, J. Muñuzuri, L. Onieva, "Viral systems: A new bio-inspired optimization approach". Computers & operations research Issue 35 (2008) pp. 2840-2860.
- [9] M. Dorigo, T. Stützle, 2004 "The ant colony optimization metaheuristic: algorithms, applications, and advances". Ant colony optimization, M. Dorigo, T. Stützle, MIT press, 2004, pp.25-64.
- [10] F. Glover, J. Kelly, M. Laguna, "Genetic algorithms and tabu search: hybrids for optimization". computer Operation Research vol.22 No. 1, 1995 pp. 111-134
- [11] M. Hirsch, C. Meneses, P. Pardalos, M. Resende, "Global optimization by continuous GRASP". Optimization letters, vol. 1 issue 2, 2006 pp.. 201-212.
- [12] N. Khilwani, A. Prakashm, R. Shankar, M. Tiwari, M. "Fast clonal algorithm". Engineering Application of Artificial Intelligence, Vol. 21, Issue 1, 2007, pp. 106-128
- [13] P. Musilek, A. Lau, M. Reformat, L. Wyard-scott, "Immune programming". Information Sciences vol 176, 2006, pp. 972-1002

Referencias

- [14] L. Yiqing, Y. Xigang, L. Yongjian, “An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints”. computer and chemical engineering Vol. 31 2007 pp. 153-162.
- [15] Z. Michalewicz, M. Schmidt, “Evolutionary Algorithms and constrained optimization” Evolutionary Optimization Kluwer’s international series, 2002, pp 58-86.
- [16] L. N. De Castro, “Fundamentals of natural computing: an overview”, physics of life reviews, vol. 4 n.1, 2007, pp. 1-36.
- [17] C.A., Coello, “An introduction to evolutionary algorithms and their applications.” Proceeding of Internacional Symposium and school of advance distributed systems, Springe-Verlay lecture notes in computer science Vol. 3563, 2005, pp. 425-442.
- [18] S.D. Muller, J. Marchetto, S. Airaghi, P. Kournoutsakos, “Optimization Algorithms based on a Model of Bacterial Chemotaxis”, proceeding of 6th Int. Conf. Simulation of Adaptive Behavior: From Animals to Animats, Paris, 2000, pp. 375-384.
- [19] Q.H. Wu, Y.J. Cao, “Stochastic Optimization of Control Parameters in Genetic Algorithms”, Proceedings of congress of evolutionary computation, 1997, pp. 77-80.
- [20] H.G., Beyer, H.P., Schwefel, “Evolution strategies”, Natural computing, vol. 1, N°. 1, 2002, pp. 3-52.
- [21] Goldberg, D., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Boston, 1989.
- [22] A.E. Eiben, J. E. Smith, Introduction to evolutionary computing, Springer, Verlag, 2003.
- [23] L. dos Santos A. A., Rodrigues, “Algoritmos evolutivos em identificação e controle de processos: uma visão integrada e perspectivas”, SBA Controle & Automação, Vol. 10m no. 01, 1999, pp.13-30.
- [24] V. Feoktistov, Differential evolution, Springer, U.S. 2006.

Referencias

- [25] P., Moscato, C. Cotta, “A introduction to memetic algoritms”, Revista Iberoamericana de Inteligencia Artificial, No.19, 2003, pp. 131-148.
- [26] Q. Nguyen, S. Ong, N. Krasnogor, A study on de designed issues of memetic algorithm. proceedings of IEEE Conference of evolutionary computation 2007. pp. 2390-2397
- [27] E. Crespo, R. Martí, J. Pacheco, Procedimientos Metaheurísticos En Economía Y Empresa, Tirant Lo Blanch, España, 2007.
- [28] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, “Optimization by Simulated Annealing”, Science, New Series, Vol. 220, No. 4598, 1983, pp. 671-680.
- [29] R. Martí, M. Laguna, “Scatter Search: Basic Design and Advanced Strategies”, Revista Iberoamericana de Inteligencia Artificial No.19, 2003, pp. 123-130.
- [30] D. Corne, M. Dorigo and F. Glover, New Ideas in Optimization, McGraw Hill, Maidenhead, 1999.
- [31] M. Dorigo, C., Blum, “Ant colony optimization theory: A survey”, Theoretical Computer Science, Vol. 344, Issues 2-3, 2005, pp. 243-278.
- [32] L. N. de Castro, J. I. Timmis, “Artificial immune systems as a novel soft computing paradigm”, Soft Computing Issue 7, 2003, pp. 526–544.
- [33] M., Resende, C. C., Ribeiro, Greedy randomized adaptive search procedures AT&T Labs Research Technical Report TD-53RSJY, 2002.

Aplicaciones

- Telecomunicaciones
 - Rutas de llamadas
 - Asignación de caminos
 - Diseño de redes para servicios
- Producción, Inventarios e Inversión
 - Fabricación flexible
 - Planificación de inventarios
- Rutas
 - Rutas de vehículos
 - Secuenciación de vagones
- Optimización Combinatoria en general
 - Programación 1-0
 - Programación No lineal No convexa
 - Optimización entera Mixta

ALGORITMOS GENETICOS ***(Genetic Algorithms GA)***

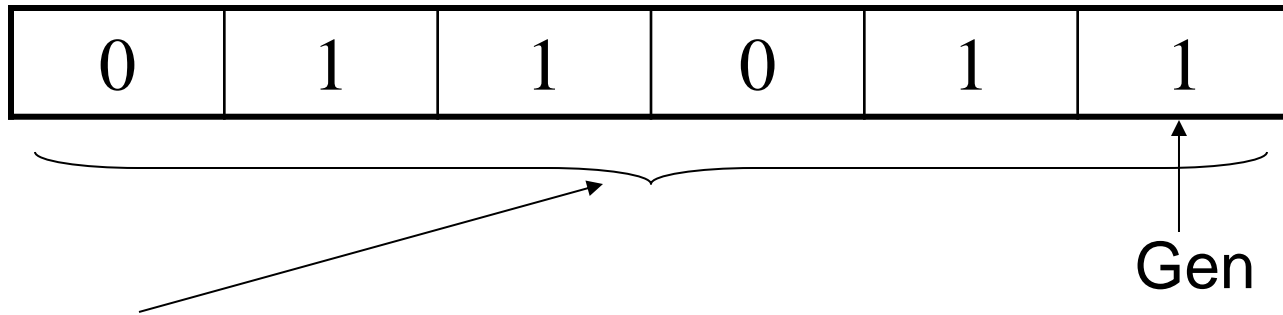
Algoritmos Genéticos

Usan el mecanismo de la **selección natural** para buscar a través del espacio de decisión, soluciones óptimas.

Buenas características se **propagan** y se mejoran de generación en generación.
Malas características desaparecen genéticamente.

La técnica intenta que un conjunto inicial diverso **converja** a una única solución

¿Cómo representa cada solución?



Cadena o cromosoma

Cadena o cromosoma: conjunto de genes que representa una solución

Gen: representan a las variables de decisión. Una variable de decisión puede ser representada por un gen o por un grupo de genes.

Los genes **controlan la herencia** de una o varias de las características de las cadenas

Longitud de la cadena: Número de genes de la cadena

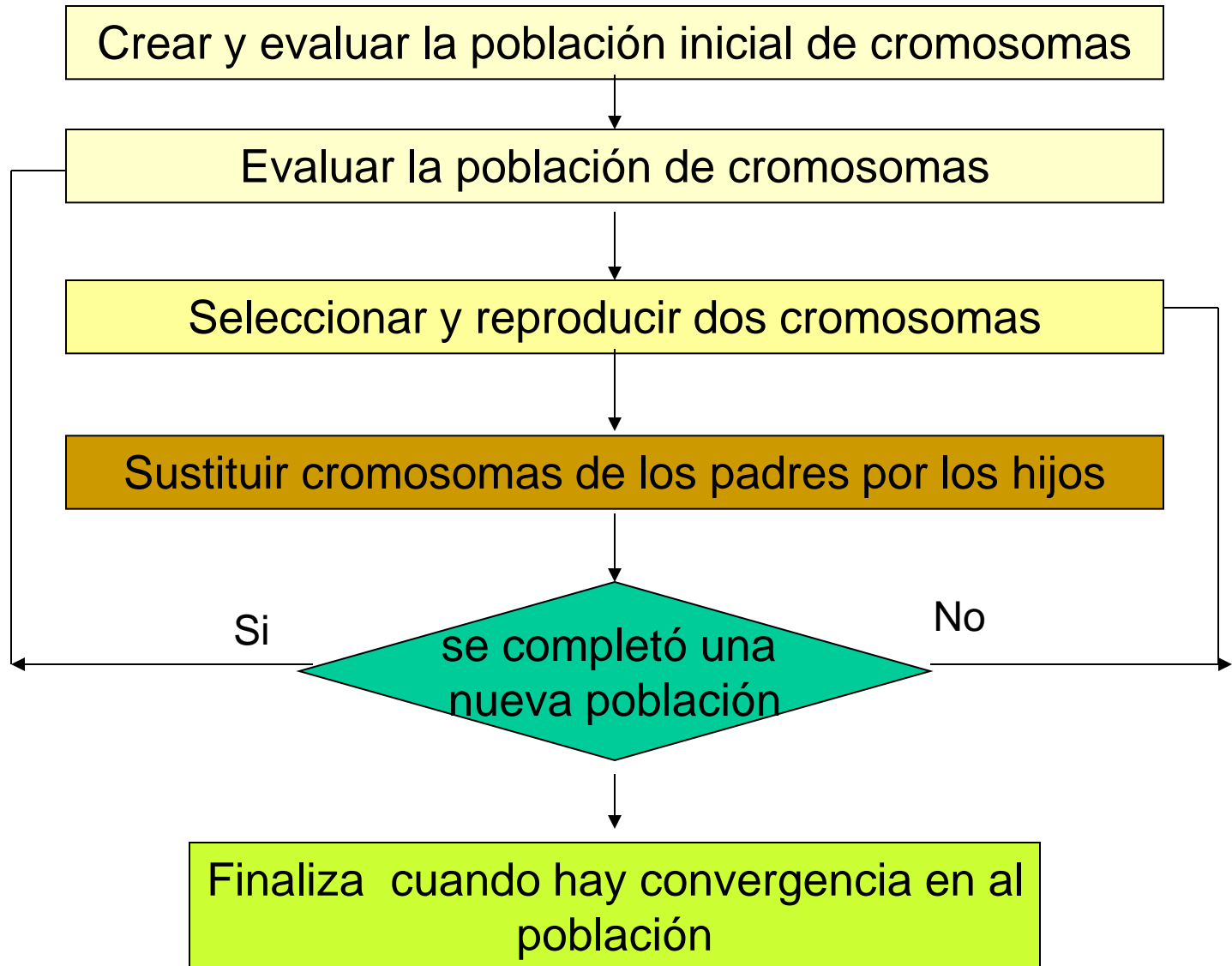
Función de Aptitud: Mide el desempeño de una solución respecto al Objetivo que se desea optimizar

Por ejemplo $Z = 4x_1^2 + 6(x_2 + x_3)^{1/3} + x_4 + x_5x_6 = 8.56$

Proceso GA

1. Crea una población inicial aleatoria
2. En cada iteración repite los siguientes pasos:
 - A. Evalúa cada individuo de la población
 - B. Con base en su aptitud, selecciona individuos de la población actual para “procrear”
 - C. Mediante técnicas genéticas crea hijos a partir de los padres seleccionados, creando una nueva población (nueva generación).
 - D. Evalúa si hay convergencia, o si ya se cumplieron N iteraciones para parar o seguir el proceso

Esquema Básico de los AG



1. Población inicial aleatoria

0	1	1	0	1	0
---	---	---	---	---	---

1	0	1	0	1	1
---	---	---	---	---	---

0	1	0	0	1	0
---	---	---	---	---	---

0	1	1	0	1	1
---	---	---	---	---	---

1	1	1	0	1	1
---	---	---	---	---	---

A. Evalúa cada individuo de la población

Lo hace mediante la **función de Aptitud**: Mide el desempeño de una solución respecto al Objetivo que se desea optimizar

Por ejemplo $Z = 4x_1^2 + 6(x_2 + x_3)^{1/3} + x_4 + x_5x_6 = 8.56$

Si el individuo (cadena o cromosoma) no es factible, se puede penalizar en su función de aptitud.

La *penalización* depende del grado de violación de las restricciones

1. Población inicial aleatoria

0	1	1	0	1	0
---	---	---	---	---	---

$Z=7.55$

1	0	1	0	1	1
---	---	---	---	---	---

$Z=11$

0	1	0	0	1	0
---	---	---	---	---	---

$Z=6$

0	1	1	0	1	1
---	---	---	---	---	---

$Z=8.56$

1	1	1	0	1	1
---	---	---	---	---	---

$Z=12.56$

B. Selección

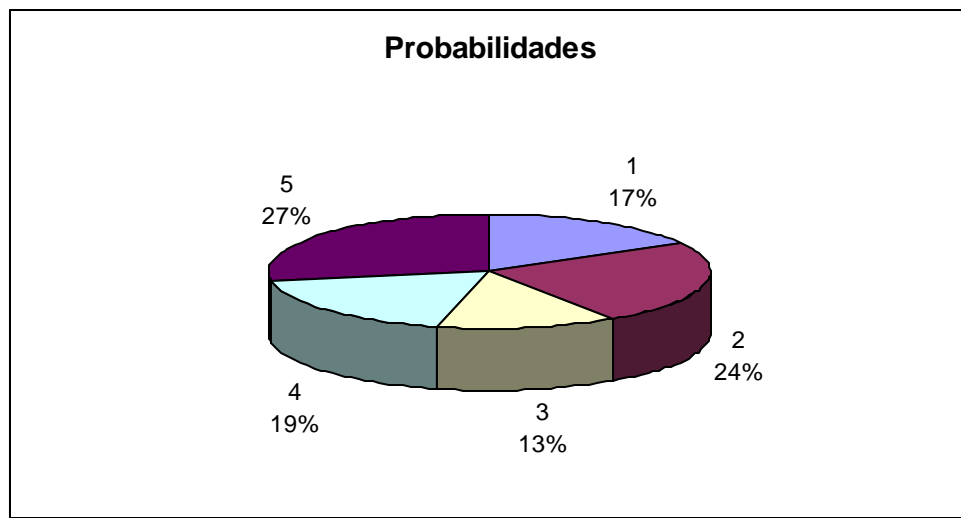
Los Individuos son seleccionados de acuerdo a su evaluación en la función objetivo (aptitud). Las más aptas tienen mayor probabilidad a contribuir genéticamente en la siguiente generación (se simula selección natural)

Puede ser por:

- Ruleta:** A cada individuo se le asigna una porción de una ruleta cuya área es 1. La porción asignada depende de su función de aptitud. Se usan números aleatorios distribuidos uniformemente para elegir cadenas
- Torneo:** Toma un cierto número de individuos y aquel que tenga el mejor valor de aptitud es seleccionado para procrear.

Ruleta:

Cadena	Aptitud	Probabilidad
1	7.55	0.17
2	11.00	0.24
3	6.00	0.13
4	8.56	0.19
5	12.56	0.28
Total	45.67	1.00

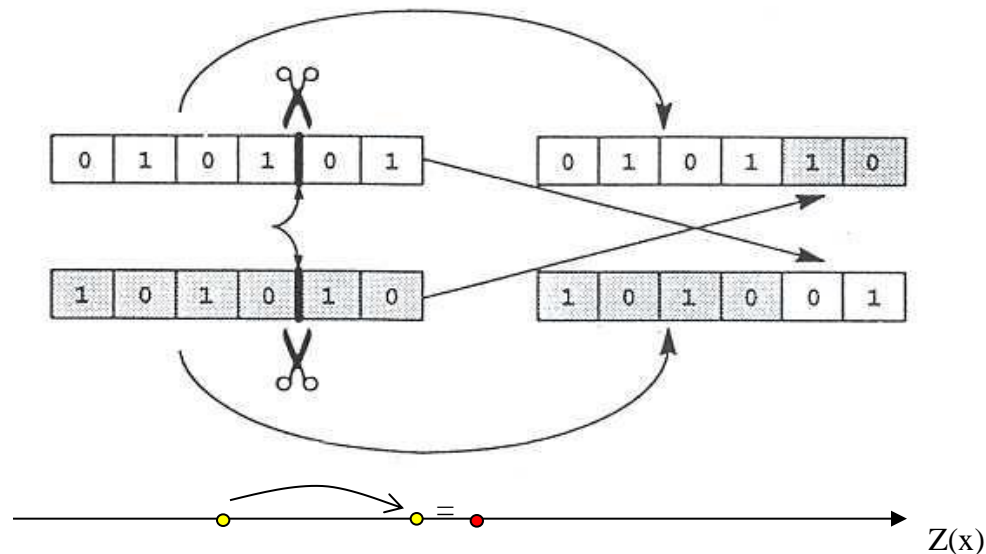


C. Generación de nuevos individuos para la próxima generación.

Operadores genéticos: cruzamiento y mutación

• **Cruzamiento**. Genera dos individuos para la nueva población a partir del cruce de dos individuos seleccionados de la generación anterior

Determina una posición de cruce aleatoria e intercambia las cadenas entre la posición inicial y el punto de cruce y el punto de cruce y la posición final

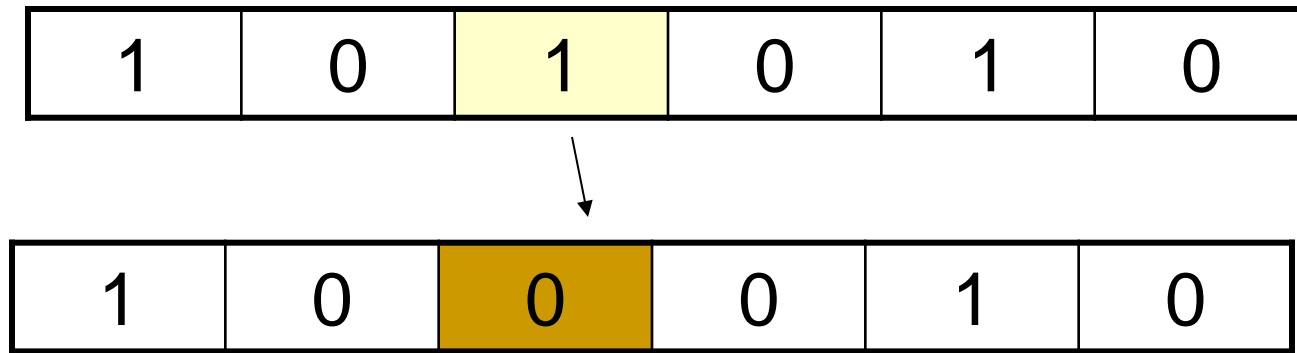


Cruce:

tiene una alta probabilidad de ser utilizado y es considerado como el más importante dentro de los AG.

Permite la generación de nuevos individuos tomando características de individuos padres.

Mutación: Opera sobre un solo individuo, determina una posición y la invierte con cierta probabilidad



No se utiliza siempre, sino que se utiliza con una baja probabilidad (del orden de 5% o menor)

permite introducir nueva información no presente en la población.

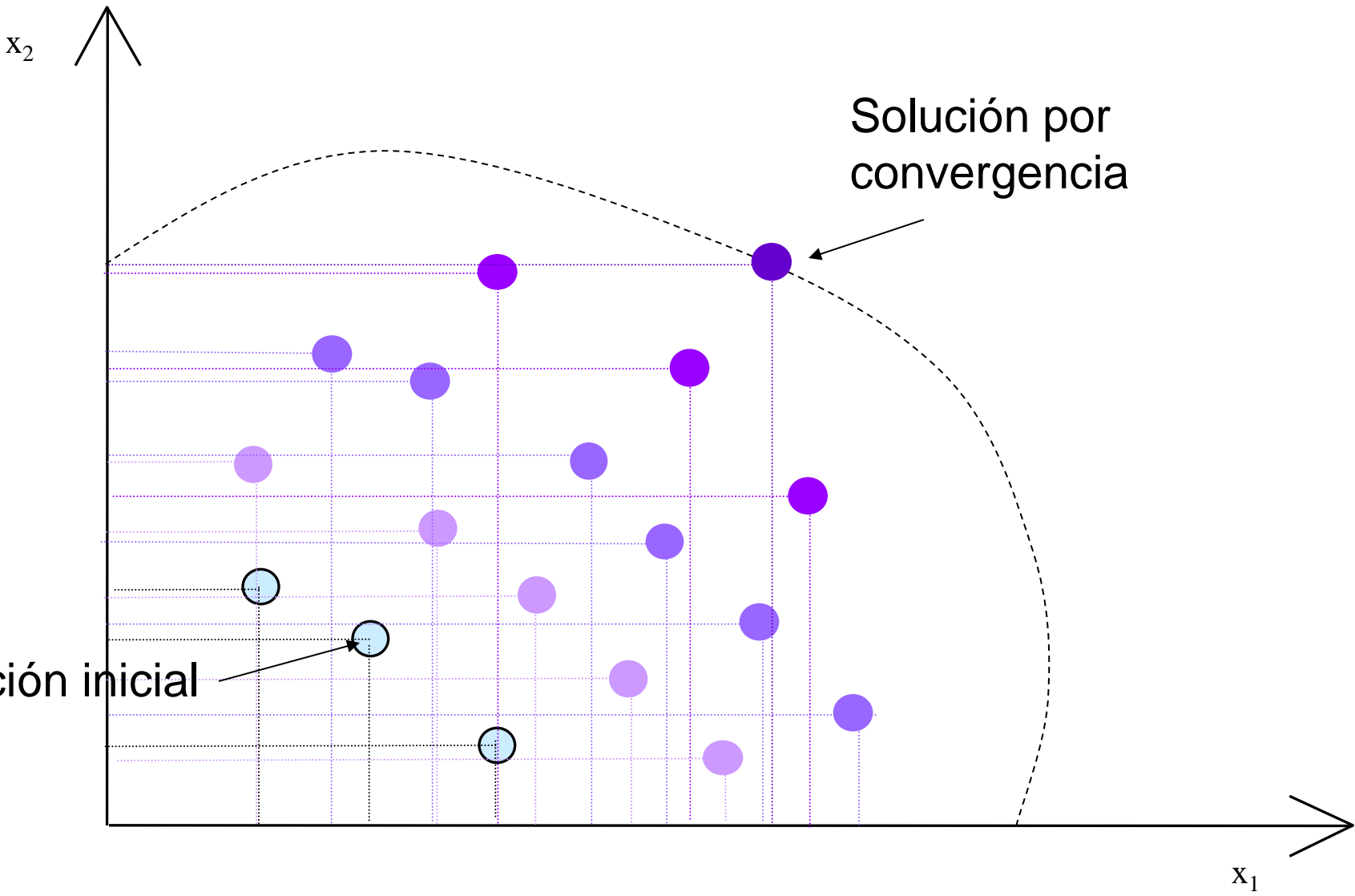
Permite salir de máximos locales.

Se generan tantas cadenas como se requieran para completar una nueva población:

Se espera que esta nueva población supere **estadísticamente** a la anterior.

D. Se crean nuevas generaciones hasta que la población converja, es decir todas las cadenas de la población tenga igual función de aptitud.

En la realidad esto es muy difícil de conseguir. Por lo que se en la práctica define un número fijo N y la solución óptima (en realidad pseudo-óptima) es la correspondiente a la cadena con mejor aptitud.



Población inicial

Solución por convergencia

Lo que el algoritmo hace es encontrar la combinación óptima de genes, utilizando para ello la experiencia adquirida en combinaciones anteriores almacenada en unos patrones genéticos.

Penalización

Castiga la probabilidad de selección para el cruce a soluciones no factibles

Para Restricciones de la forma $g_j(\mathbf{x}) < B_j$

Se calcula: $k_j = \max\{0, g_j(\mathbf{x}) - B_j\}$

Homaifer, Lai y Qi

$$Z^p(\mathbf{x}) = Z(\mathbf{x}) - \sum_{j=1}^m R_{ij} k_j^2(\mathbf{x})$$

R_{ij} = Penalización por la restricción j para el rango de violación i

Joines y Houck

$$Z^p(\mathbf{x}) = Z(\mathbf{x}) - (C.t)^\alpha \sum_{i=1}^m k_j^\beta(\mathbf{x})$$

C , α y β = constantes, t =iteración

Ventajas de los GA

- Permite funciones objetivo altamente no lineales.
- La búsqueda es guiada por información de valores de la función objetivo, que no necesita ser continua, diferenciable o unimodal.
- Buscan en una población y no en un único punto. Esto incrementa la probabilidad de que no quede atrapado en un óptimo local.
- Permite trabajar en paralelo .

Ventajas de los GA

- Permite buscar soluciones en espacios de decisión multimodales complejos.
- Usa la función objetivo en si misma y no usa información derivada.
- Usa reglas probabilísticas en vez de determinísticas como la mayoría de métodos tradicionales

Ejemplo:

Se han elaborado 6 propuestas de proyectos de generación de energía en la cuenca B muy importantes, y se cuenta sólo con cierto capital.

Cada propuesta i se caracteriza por:

- C_i costo
- G_i la cantidad de energía que genera anualmente.

Se requiere elegir, cuales alternativas implementar para Maximizar la generación anual de energía (en GWh)

$$Z(\mathbf{x}) = \text{Max} \quad \left\{ \sum_{i=1}^n G_i x_i \right\}$$

Restricciones

El capital total de inversión no debe superar a $\$16000 \times 10^9$

Existe incompatibilidad entre los proyectos 1 y 5.

$$\sum_{i=1}^n C_i x_i \leq 16000$$

$$x_1 + x_5 \leq 1$$

$$x_j = (0,1) \quad \text{para todo } j = 1, \dots, n$$

Características de los Proyectos

	Energía generada (GWh) G_i	Area importante afectada (Ha) A_i	Inversión requerida (10⁹ \$) C_i
1	409	0	4130
2	1718	500	4210
3	2000	495	2400
4	2008	490	2420
5	2075	780	2020
6	1800	500	5200

Formulación

$$\text{Maximizar} \quad \sum_{i=1}^n G'_i x_i$$

$$\text{Sujeto a:} \quad \sum_{i=1}^n C_i x_i \leq 16000$$

$$x_1 + x_5 \leq 1$$

$$x_j = (0,1)$$

Parámetros

No de Iteraciones 50

Población: 20

Probabilidad de cruce: 0.98

Probabilidad de mutación: 0.01

Población inicial

i	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	Z GWh	Costo	Incom patibili dad	Prob.	Prob. Acumu lada
1	0	0	1	1	1	0	6083	6840	1	0.065	0.065
2	1	0	0	1	0	1	4417	11750	1	0.045	0.110
3	1	0	0	1	0	1	4417	11750	1	0.045	0.156
4	1	1	0	1	0	0	4135	10760	1	0.044	0.200
5	0	1	0	0	1	1	5793	11430	1	0.060	0.260
6	1	0	1	0	0	0	2409	6530	1	0.026	0.286
7	0	0	0	0	1	0	2075	2020	1	0.022	0.308
8	0	0	1	1	0	1	6008	10020	0	0.062	0.370
9	1	1	1	1	0	0	6135	13160	1	0.066	0.436
10	1	1	1	0	0	0	4127	10740	1	0.044	0.480
11	0	0	1	1	0	0	4008	4820	0	0.043	0.523
12	0	1	1	0	1	1	7793	13830	1	0.081	0.605
13	0	0	0	0	0	1	2000	5200	0	0.019	0.624
14	0	1	0	1	0	1	5726	11830	0	0.059	0.683
15	0	0	0	1	0	1	4008	7620	0	0.041	0.724
16	1	1	1	0	0	1	6127	15940	1	0.064	0.788
17	1	0	1	1	0	1	6417	14150	1	0.067	0.854
18	0	0	1	0	1	0	4075	4420	1	0.044	0.898
19	0	1	0	1	1	0	5801	8650	1	0.062	0.960
20	0	1	1	0	0	0	3718	6610	0	0.040	1.000

Selección de padres:

aleatorio: 0.56, Alternativa 12

0	1	1	0	1	1
---	---	---	---	---	---

aleatorio: 0.81, Alternativa 17

1	0	1	1	0	1
---	---	---	---	---	---

aleatorio: 0.86 < 0.98= ¡¡¡se cruzan!!!

Cada posible corte tiene igual probabilidad, 1/5

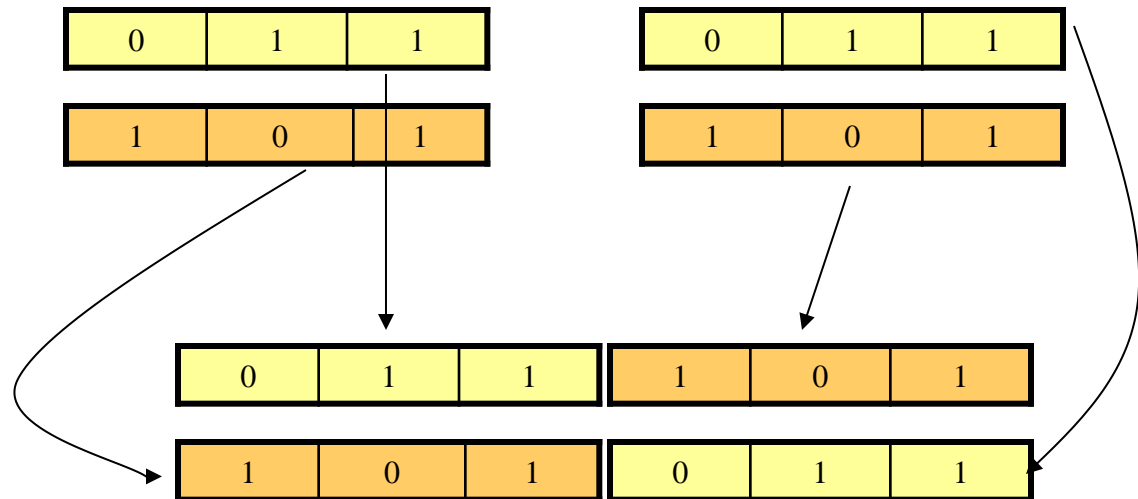
aleatorio: 0.51. \Rightarrow Punto de cruce: 3

Padres

0	1	1
1	0	1

0	1	1
1	0	1

Hijos



Mutación:

Hijo 1	0	1	1	1	0	1
Hijo 2	1	0	1	0	1	1

Para el Hijo 1

Gen 1: # aleatorio $0.58 > 0.05$, no hay mutación

Gen 2: # aleatorio $0.005 < 0.01$, ¡¡¡ hay mutación!!!.....

Supóngase que no se definieron más mutaciones:

Hijo 1	0	0	1	1	0	1
Hijo 2	1	0	1	0	1	1

Verifica si cumplen las restricciones:

Hijo 1

0	0	1	1	0	1	5808	10020	0
---	---	---	---	---	---	------	-------	---

Hijo 2

1	0	1	0	1	1	6284	13750	2
---	---	---	---	---	---	------	-------	---



No cumple
restricción
2, luego, se
penalizará
en la
función de
aptitud

Igualmente, a partir de la selección de padres y creación de hijos (cruzamiento y mutación) se genera una nueva población de 20 cadenas o individuos

En la última iteración la mejor solución fue:

0	0	1	1	1	1	7883	12040	1
---	---	---	---	---	---	------	-------	---

Es decir, Se hacen los proyectos 3, 4, 5, 6

Bibliografía GA

- David E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

Melanie Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Massachusetts, 1996.

- Zbigniew Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Second Edition, 1992.
- John H. Holland, *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, Massachusetts, Second Edition, 1992.
- David B. Fogel, *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*, The Institute of Electrical and Electronic Engineers, New York, 1995.
- Thomas Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.