



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Algoritmos Evolutivos ++

Fernando Berzal, berzal@acm.org

Algoritmos Evolutivos ++



Preservación de la diversidad en subpoblaciones

- Selección: Nichos & especiación
- Algoritmos evolutivos paralelos (islas) & celulares

Otras variantes de algoritmos evolutivos

- Sistemas clasificadores: LCS [Learning Classifier Systems]
- Evolución diferencial
- PSO [Particle Swarm Optimization]
- EDA [Estimation of Distribution Algorithms]



Nichos: Fitness sharing



IDEA: Se controla el número de individuos en cada nicho compartiendo su fitness justo antes de la selección.

Se asignan individuos a nichos en función del fitness del nicho:

$$F'(i) = \frac{F(i)}{\sum_j sh(d(i, j))} \quad sh(d) = \begin{cases} 1 - (d/\sigma_{share})^\alpha & \text{if } d \leq \sigma_{share}, \\ 0 & \text{otherwise.} \end{cases}$$

Uso con selección proporcional (FPS).

D.E. Goldberg & J. Richardson: "Genetic algorithms with sharing for multimodal function optimization." ICGA'1987



Nichos: Crowding



IDEA: Preservar la diversidad asegurándonos de que los nuevos individuos reemplazan a miembros similares de la población.

ESQUEMA ORIGINAL:

Algoritmo estacionario en el que los descendientes reemplazan a sus padres más similares.

K.A. De Jong: "An Analysis of the Behaviour of a Class of Genetic Adaptive Systems." PhD thesis, University of Michigan, 1975.

S.W. Mahfoud. Crowding and preselection revisited. PPSN'1992

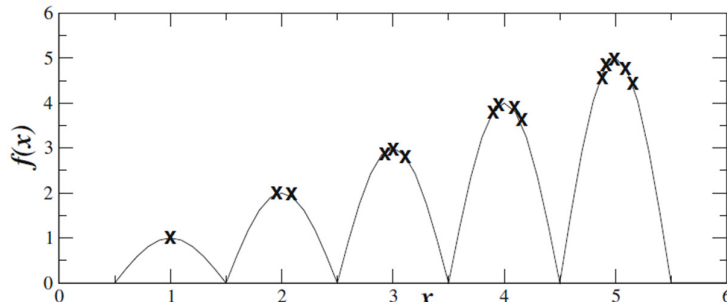


Nichos

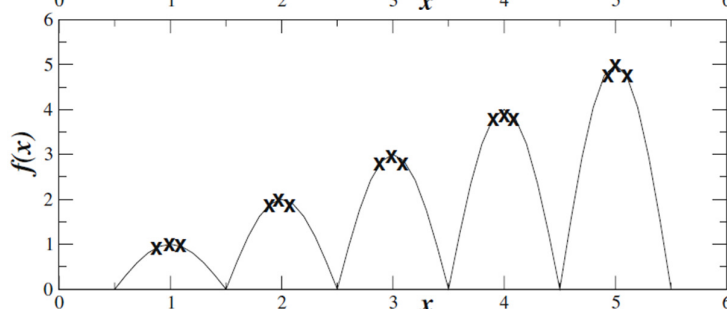


RESULTADO:

Se preservan subpoblaciones en nichos...



Fitness sharing



Crowding



Especiación



IDEA: Restricciones sobre la reproducción basadas en algún aspecto de las soluciones candidatas (o su genoma), que se asocia a especies diferentes.

MECANISMO: Población con múltiples especies, en la cual los individuos sólo se reproducen con miembros de su misma (o similar) especie.

ESQUEMAS

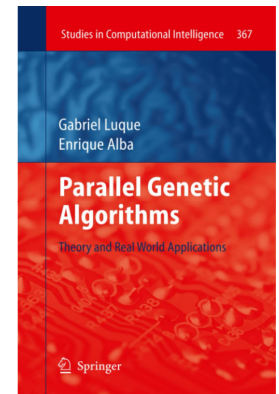
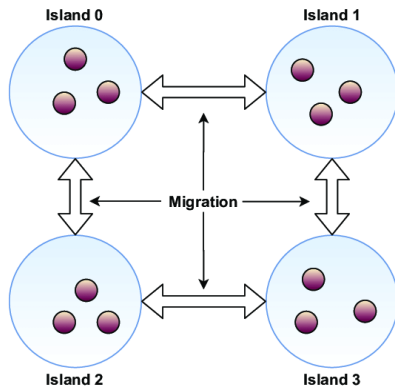
- Emparejamiento restringido por fenotipos (Deb)
K. Deb & D.E. Goldberg: "An investigation of niche and species formation in genetic function optimization." ICGA'1989
- Etiquetas asociadas al genotipo
W.M. Spears: "Simple subpopulation schemes", CEP'1994



AE paralelos: Modelos de islas

IDEA:

Varias poblaciones evolucionan en tándem.



Modelos de islas populares desde finales de los años 80 (popularización de los sistemas paralelos de cómputo)...

Gabriel Luque & Enrique Alba (Universidad de Málaga):
"Parallel Genetic Algorithms,"
Studies in Computational Intelligence, 367. Springer, 2011.



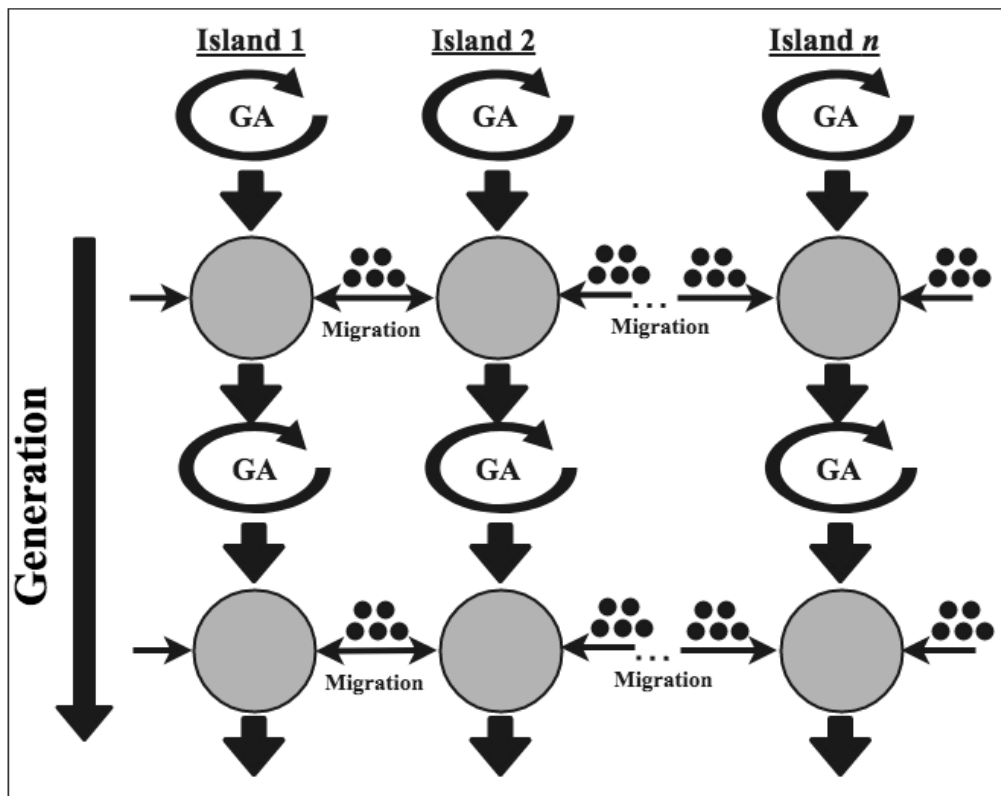
AE paralelos: Modelos de islas

Comunicación entre islas

- Topología dependiente a menudo de la arquitectura del sistema paralelo (anillo, toro, hipercubo...).
- Tras un número fijo de generaciones (una **época**), se intercambian individuos entre poblaciones vecinas (una **migración**).
- Explotación entre migraciones (islas evolucionando independientemente), exploración al inyectar nuevos individuos en una subpoblación.



AE paralelos: Modelos de islas



AE celulares

IDEA:

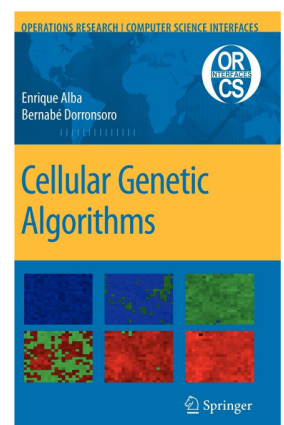
Una única población se subdivide en subpoblaciones solapadas (demes).

MECANISMO:

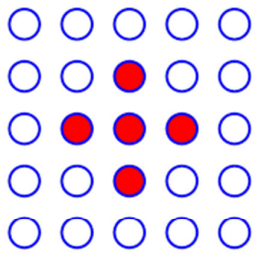
Los individuos sólo compiten y se reproducen con sus vecinos.

IMPLEMENTACIÓN paralela en FPGAs o GPUs.

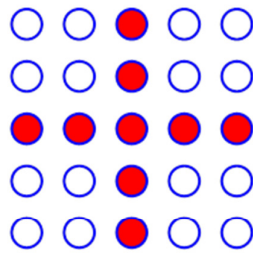
Enrique Alba (Universidad de Málaga) & Bernabé Dorronsoro (Universidad de Cádiz): "Cellular Genetic Algorithms." Computational Intelligence and Complexity. Springer, 2008



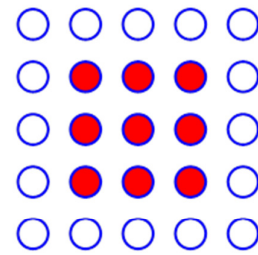
AE celulares



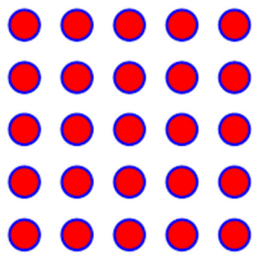
L5



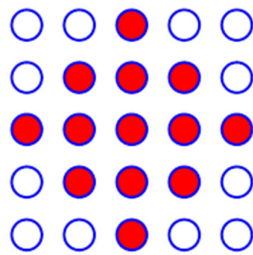
L9



C9



C25



D13



https://en.wikipedia.org/wiki/Cellular_evolutionary_algorithm



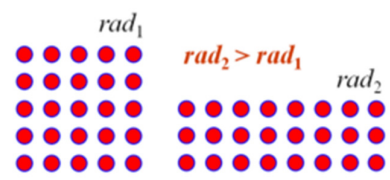
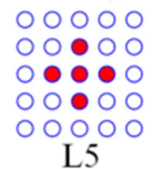
AE celulares



$$rad = \sqrt{\frac{\sum(x_i - \bar{x})^2 + \sum(y_i - \bar{y})^2}{n^*}} \quad \bar{x} = \frac{\sum_{i=1}^{n^*} x_i}{n^*} \quad \bar{y} = \frac{\sum_{i=1}^{n^*} y_i}{n^*}$$

$$ratio_{cGA} = \frac{rad_{neighborhood}}{rad_{grid-topology}}$$

$$rad_{L5} = \sqrt{\frac{2+2}{3+2}} = 0.8944$$



La razón entre el radio del vecindario y la topología de la población define la capacidad de exploración/explotación del algoritmo evolutivo celular [cEA]





- Sistemas clasificadores:
LCS [Learning Classifier Systems]
- Evolución diferencial
- PSO [Particle Swarm Optimization]
- EDA [Estimation of Distribution Algorithms]



Sistemas clasificadores



LCS [Learning Classifier Systems]

Evolucionan conjuntos de reglas:

- **Enfoque Michigan**
Cada individuo es una regla.
Similitudes con el aprendizaje por refuerzo.
- **Enfoque Pittsburgh**
Cada individuo es un conjunto completo de reglas.
Más parecido a la programación genética.



Sistemas clasificadores



LCS [Learning Classifier Systems]

Estilo Michigan

Representation	tuple of {condition:action:payoff,accuracy} conditions use {0,1,#} alphabet
Recombination	One-point crossover on conditions/actions
Mutation	Binary/ternary resetting as appropriate on action/conditions
Parent selection	Fitness proportional with sharing within environmental niches
Survivor selection	Stochastic, inversely related to number of rules covering same environmental niche
Fitness	Each reward received updates predicted payoff and accuracy of rules in relevant action sets by reinforcement learning.

ZCS

S.W. Wilson. ZCS: A zeroth level classifier system.
Evolutionary Computation, 2(1):1–18, 1994.

XCS

S.W. Wilson. Classifier fitness based on accuracy.
Evolutionary Computation, 3(2):149–175, 1995.



Sistemas clasificadores



LCS [Learning Classifier Systems]

Estilo Pittsburgh

- Cromosomas de longitud variable.
- Cada gen representa una regla.
- Lista ordenada de reglas (i.e. lista de decisión)



Humies Awards (for better than human performance)

Detección de cancer de próstata

X. Llorca et al.: "Towards better than human capability in diagnosing prostate cancer using infrared spectroscopic imaging", GECCO'2007

Predicción de estructura de proteínas

J. Bacardit et al.: "Krasnogor. Automated alphabet reduction method with evolutionary algorithms for protein structure prediction,"
GECCO'2007



Evolución diferencial



DE [Differential Evolution], 1995

- Heurística para “minimizar funciones posiblemente no lineales y no diferenciables en espacios continuos”

- **Mutación diferencial**

Se crea un vector mutante añadiendo una perturbación a un vector existente, donde la perturbación es una versión escalada de la diferencia entre otros dos vectores existentes en la población:

$$\mathbf{x} = \mathbf{x} + \mathbf{p}$$

$$\mathbf{p} = F (\mathbf{y} - \mathbf{z}) \quad F > 0 \text{ (factor de escala)}$$



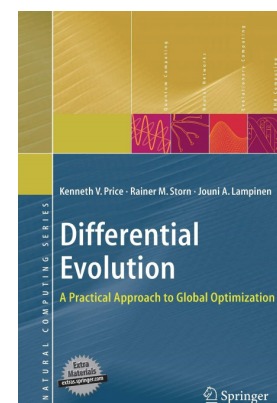
Evolución diferencial



Representation	Real-valued vectors
Recombination	Uniform crossover
Mutation	Differential mutation
Parent selection	Uniform random selection of the 3 necessary vectors
Survival selection	Deterministic elitist replacement (parent vs. child)

3 parámetros

- Factor de escala F
- Tamaño de la población (μ)
- Probabilidad de cruce



K.V. Price, R.N. Storn & J.A. Lampinen:
Differential Evolution: A Practical Approach to Global Optimization.
Natural Computing Series. Springer, 2005.





Variantes DE/a/b/c

- **Vector base a: Elitismo (a=best) vs. a=rand**
Escoger siempre el mejor individuo como base para los mutantes (sólo varían los vectores de perturbación).
- **Modelo de perturbación (b=2)**
Usar más de un vector de diferencias al definir el vector de perturbación, p.ej. 4 individuos $\mathbf{p} = F(\mathbf{y} - \mathbf{z} + \mathbf{y}' - \mathbf{z}')$
- **Operador de cruce (c=bin)**
Cruce uniforme (genera una distribución **binomial**).



PSO [Particle Swarm Optimization]



Kennedy & Eberhart, ICNN'1995

- Inspirado en el comportamiento social de las bandadas de pájaros [bird flocking] o los bancos de peces [fish schooling].
- Terminología no biológica propia de partículas físicas: posición y velocidad.
- Como en la evolución diferencial, cambia la forma de interpretar los operadores de variación:
 - PSO no utiliza recombinación.
 - PSO utiliza mutación basada en suma de vectores.



PSO [Particle Swarm Optimization]



- Cada solución candidata (\mathbf{x}) lleva consigo su propio vector de perturbación (\mathbf{p}), como las estrategias de evolución autoadaptativas: individuos $\langle \mathbf{x}, \mathbf{p} \rangle$.
- Cada miembro de la población se interpreta como una partícula en el espacio con posición y velocidad.
- La velocidad v de la partícula x se actualiza teniendo en cuenta la posición de la mejor partícula actual (y) y la de la mejor encontrada nunca (z):

$$\mathbf{v}' = w \mathbf{v} + \phi_1 U_1(\mathbf{y} - \mathbf{x}) + \phi_2 U_2(\mathbf{z} - \mathbf{x})$$



PSO [Particle Swarm Optimization]



Parámetros

- w Inercia
- ϕ_1 Tasa de aprendizaje para la influencia personal
- ϕ_2 Tasa de aprendizaje para la influencia social

U_1 y U_2 generan valores aleatorios uniformes para cada componente de los vectores de perturbación

La posición de las partículas se actualiza de acuerdo a su velocidad:

$$\mathbf{x}' = \mathbf{x} + \mathbf{v}'$$

$$\mathbf{v}' = w \mathbf{v} + \phi_1 U_1(\mathbf{y} - \mathbf{x}) + \phi_2 U_2(\mathbf{z} - \mathbf{x})$$



EDA [Estimation of Distribution Algorithms]

Reemplazan los operadores estándar de variación (recombinación y mutación) por un proceso de 3 etapas:

- Selección de un modelo
- Ajuste de los parámetros de un modelo
- Muestreo del modelo

Se selecciona un modelo gráfico probabilístico (red bayesiana o mezcla de gaussianas) que representa el estado actual de la búsqueda en términos de las dependencias entre variables que describen una solución candidata y se emplea como los modelos generativos en Deep Learning.



EDA [Estimation of Distribution Algorithms]

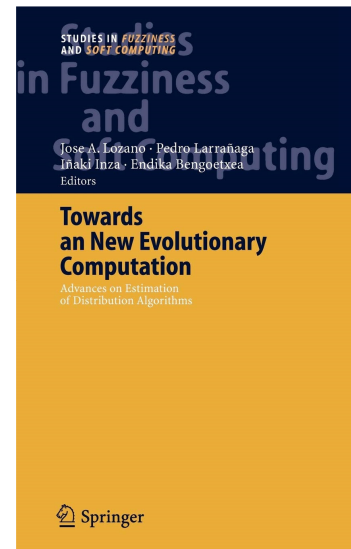
En el contexto de genética (y computación evolutiva), la selección del modelo adecuado se conoce con el término “**linkage learning**”:

- **PBIL** [Population-Based Incremental Learning]
Variables independientes (CMU'1995)
- **MIMIC** (NIPS'1997) & **BMDA** (ASCEM'1999)
Interacciones entre pares de variables
- **BOA** [Bayesian Optimization Algorithm]
Árboles de tamaño máximo (GECCO'1999)



EDA [Estimation of Distribution Algorithms]

Extensiones y comparaciones...



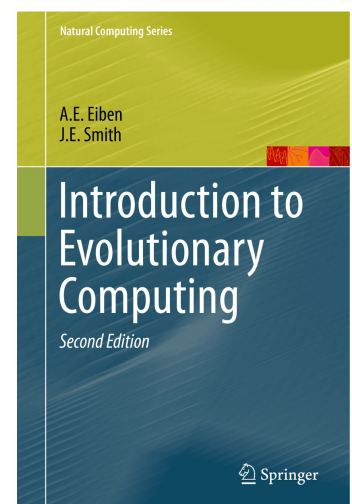
José A. Lozano, Pedro Larrañaga, Iñaki Inza & Endika Bengoetxea:
"Towards a New Evolutionary Computation: Advances on Estimation
of Distribution Algorithms," Studies in Fuzziness and Soft Computing
192, 2006



Bibliografía

Lecturas recomendadas

- A.E. Eiben & J.E. Smith:
**Introduction to
Evolutionary Computing**
Springer, 2nd edition, 2015
ISBN 3662448734
<http://www.evolutionarycomputation.org/>

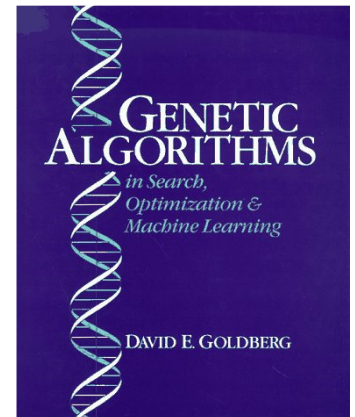
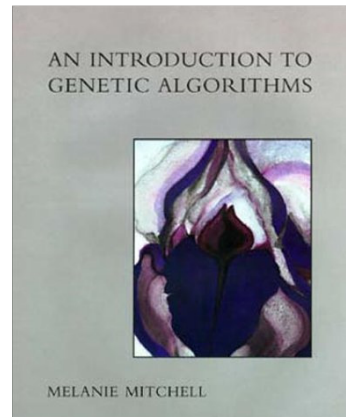


Bibliografía



Bibliografía complementaria

- Melanie Mitchell:
An Introduction to Genetic Algorithms
MIT Press, 1996.
ISBN 0262133164



- David E. Goldberg:
Genetic Algorithms in Search, Optimization & Machine Learning.
Addison-Wesley, 1989.
ISBN 0201157675



Bibliografía



Bibliografía complementaria

- John R. Koza:
**Genetic Programming:
On the Programming of Computers
by Means of Natural Selection.**
MIT Press, 1992.
ISBN 0-262-11170-5

