



METAHEURÍSTICAS

2014 - 2015

- Tema 1. Introducción a las Metaheurísticas
- Tema 2. Algoritmos de Búsqueda Local Básicos
- Tema 3. Métodos Basados en Trayectorias Simples
- Tema 4. Métodos Basados en Trayectorias Múltiples
- Tema 5. Metaheurísticas Basadas en Poblaciones
- Tema 6. Metaheurísticas Basadas en Adaptación Social
- Tema 7. Metaheurísticas Híbridas
- Tema 8. Metaheurísticas Paralelas

METAHEURÍSTICAS

TEMA 7. METAHEURÍSTICAS HIBRIDAS

1. ALGORITMOS MEMÉTICOS

2. BÚSQUEDA DISPERSA (SCATTER SEARCH)

Bibliografía

P. Moscato, C. Cotta, "A Gentle Introduction to Memetic Algorithms", . In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 105-144, Kluwer, Boston MA, 2003.

P. Moscato, C. Cotta, Una Introducción a los Algoritmos Meméticos, Inteligencia Artificial 19 (2003) 131-148.

M. Laguna, R. Martí. Scatter Search. Kluwer, 2002

R. Martí, M. Laguna. Scatter Search: Diseño Básico y Estrategias Avanzadas. Revista Iberoamericana de Inteligencia Artificial 19 (2003) 123-130

F. Glover, M. Laguna, R. Martí. Chapter 1: Scatter Search and Path Relinking: Advances and Applications. In: F. Glover, G.A. Kochenberber, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 1-35.

ALGORITMOS MEMÉTICOS

SUMARIO

- **¿Qué es un Algoritmo Memético?**
- **¿Por qué hibridar? Los límites de los Algoritmos Evolutivos**
- **Algoritmos Meméticos: Introducción**
- **Sobre el Diseño de Algoritmos Meméticos**
- **Algoritmos Meméticos: Estudios Recientes**
- **Conclusiones**

SUMARIO **ALGORITMOS MEMÉTICOS**

- **¿Qué es un Algoritmo Memético?**
- **¿Por qué hibridar? Los límites de los Algoritmos Evolutivos**
- **Algoritmos Meméticos: Introducción**
- **Sobre el Diseño de Algoritmos Meméticos**
- **Algoritmos Meméticos: Estudios Recientes**
- **Conclusiones**

¿Qué es un Algoritmo Memético?

Algoritmo basado en la evolución de poblaciones que para realizar búsqueda heurística intenta utilizar todo el conocimiento sobre el problema (usualmente conocimiento en términos de algoritmos específicos de búsqueda local para el problema)

¿Por qué esta hibridación?

SUMARIO **ALGORITMOS MEMÉTICOS**

- **¿Qué es un Algoritmo Memético?**
- **¿Por qué hibridar? Los límites de los Algoritmos Evolutivos**
- **Algoritmos Meméticos: Introducción**
- **Sobre el Diseño de Algoritmos Meméticos**
- **Algoritmos Meméticos: Estudios Recientes**
- **Conclusiones**

¿Por qué hibridar?

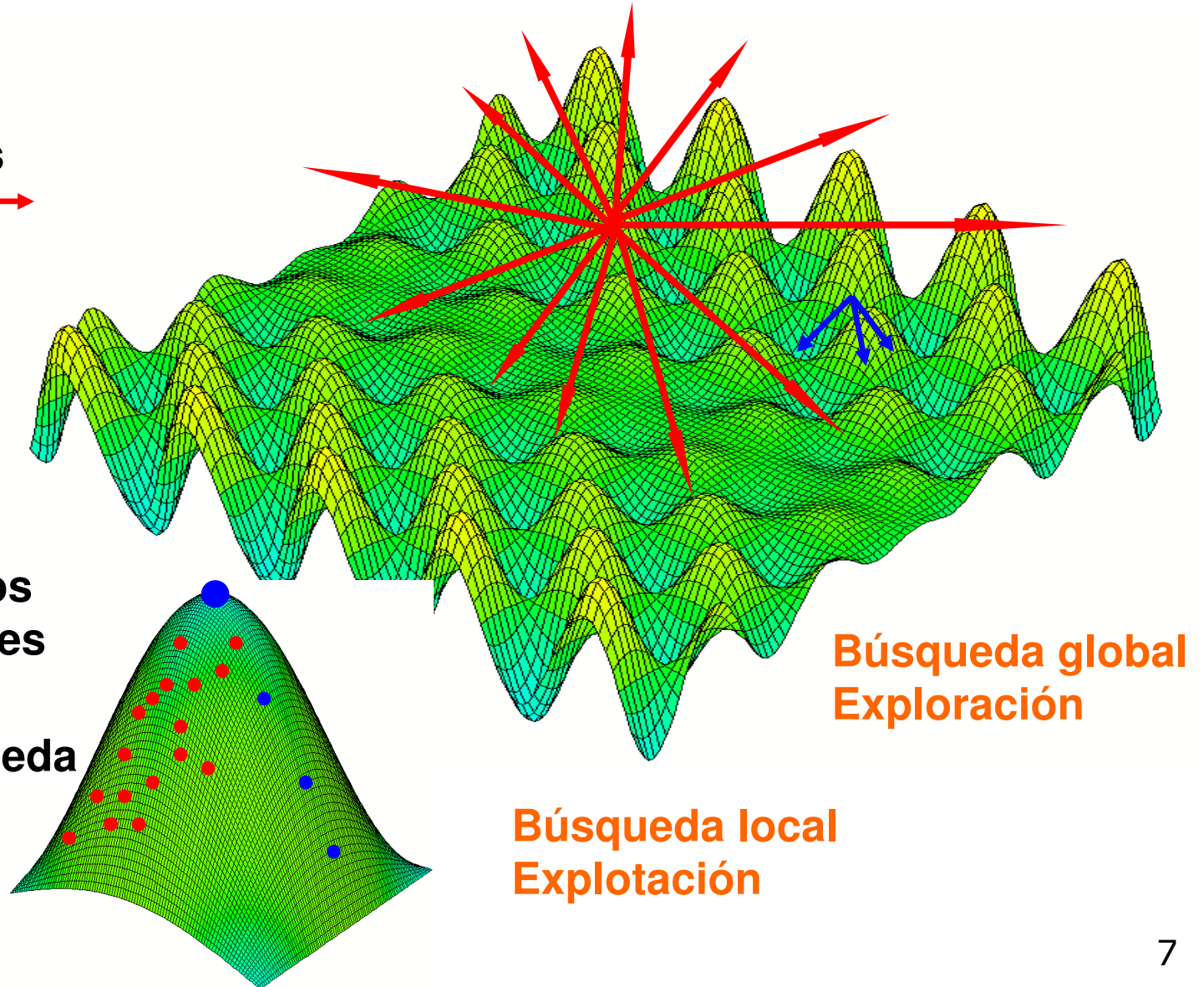
Los límites de los AEs

□ Algoritmos evolutivos
son buenos
exploradores →

□ Algoritmos de
búsqueda local son
malos exploradores →

□ Algoritmos evolutivos
son malos explotadores ●

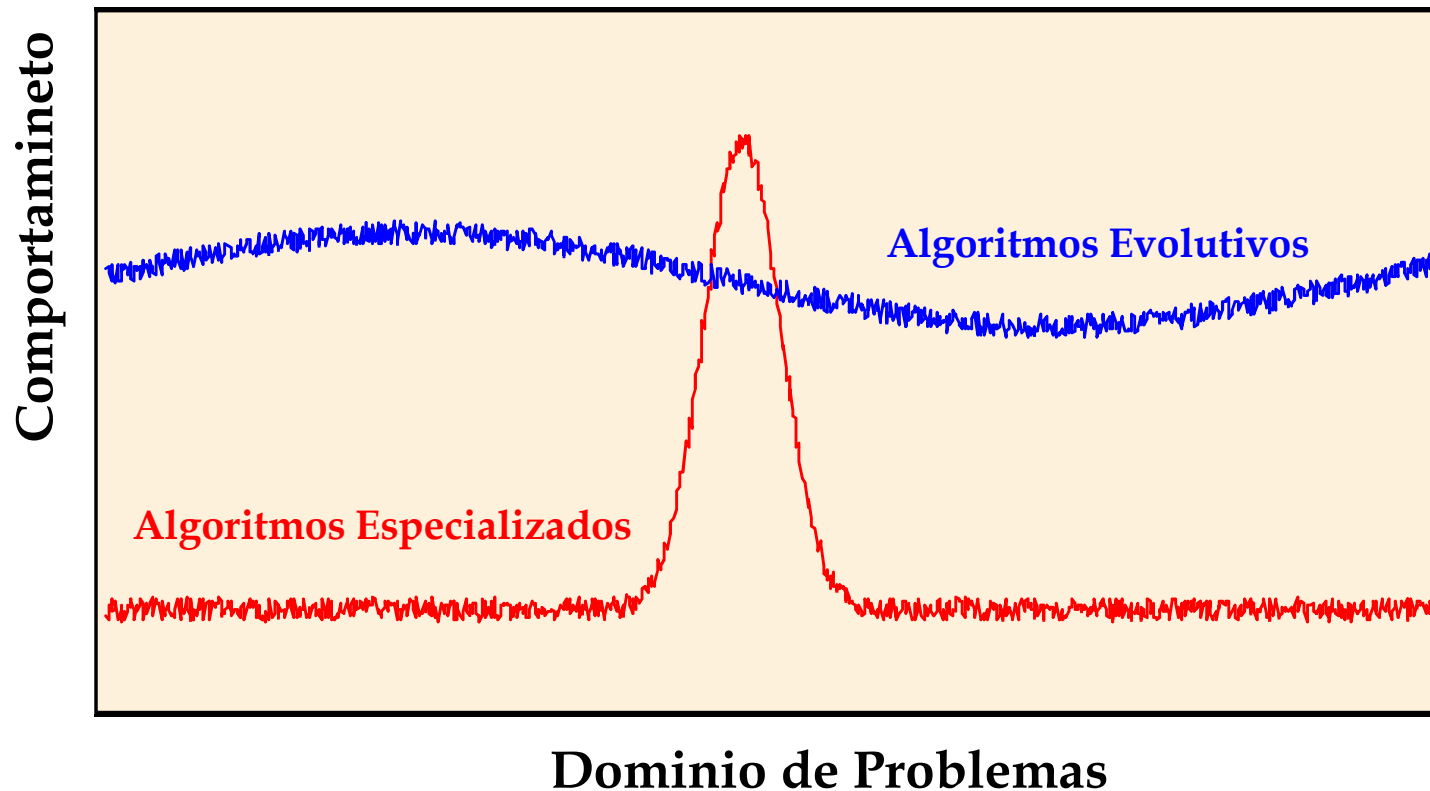
□ Algoritmos de búsqueda
local son buenos
explotadores ●



¿Por qué hibridar?

Los límites de los AEs

Sobre el Comportamiento de los Algoritmos Evolutivos



¿Por qué hibridar?

Los límites de los AEs

No Free Lunch Theorem (1995):

"...for any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class." **Wolpert and Macready (1997)**

$$\sum_f E(\vec{c} / f, m, a) = \sum_f E(\vec{c} / f, m, b)$$



David. H. Wolpert



William G. Macready

No free lunch theorems for optimization

Wolpert, D.H.; Macready, W.G.;

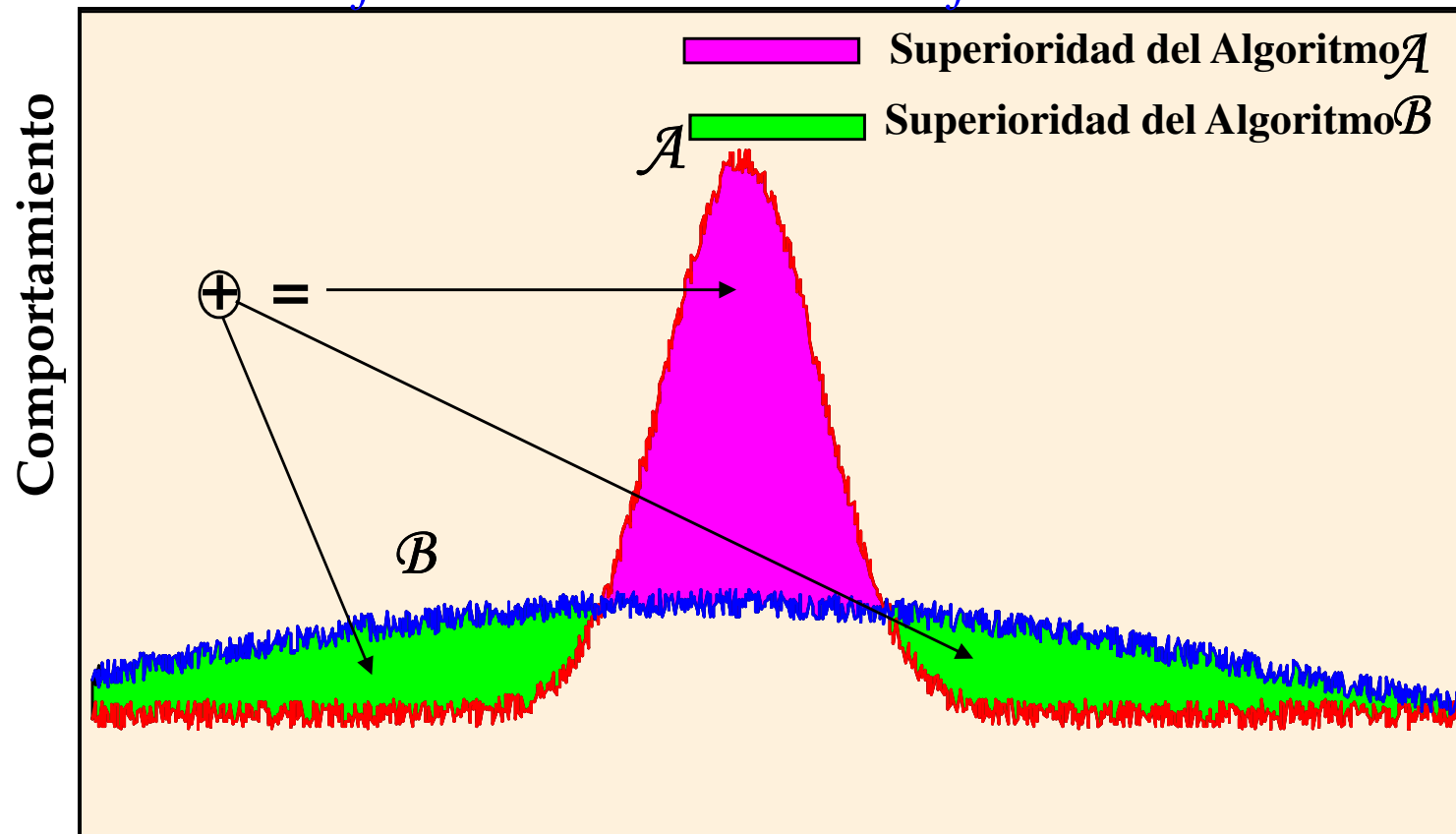
Evolutionary Computation, IEEE Transactions on 1:1, April 1997, 67 – 82

¿Por qué hibridar?

Los límites de los AEs

Implicaciones
de NFL (I)

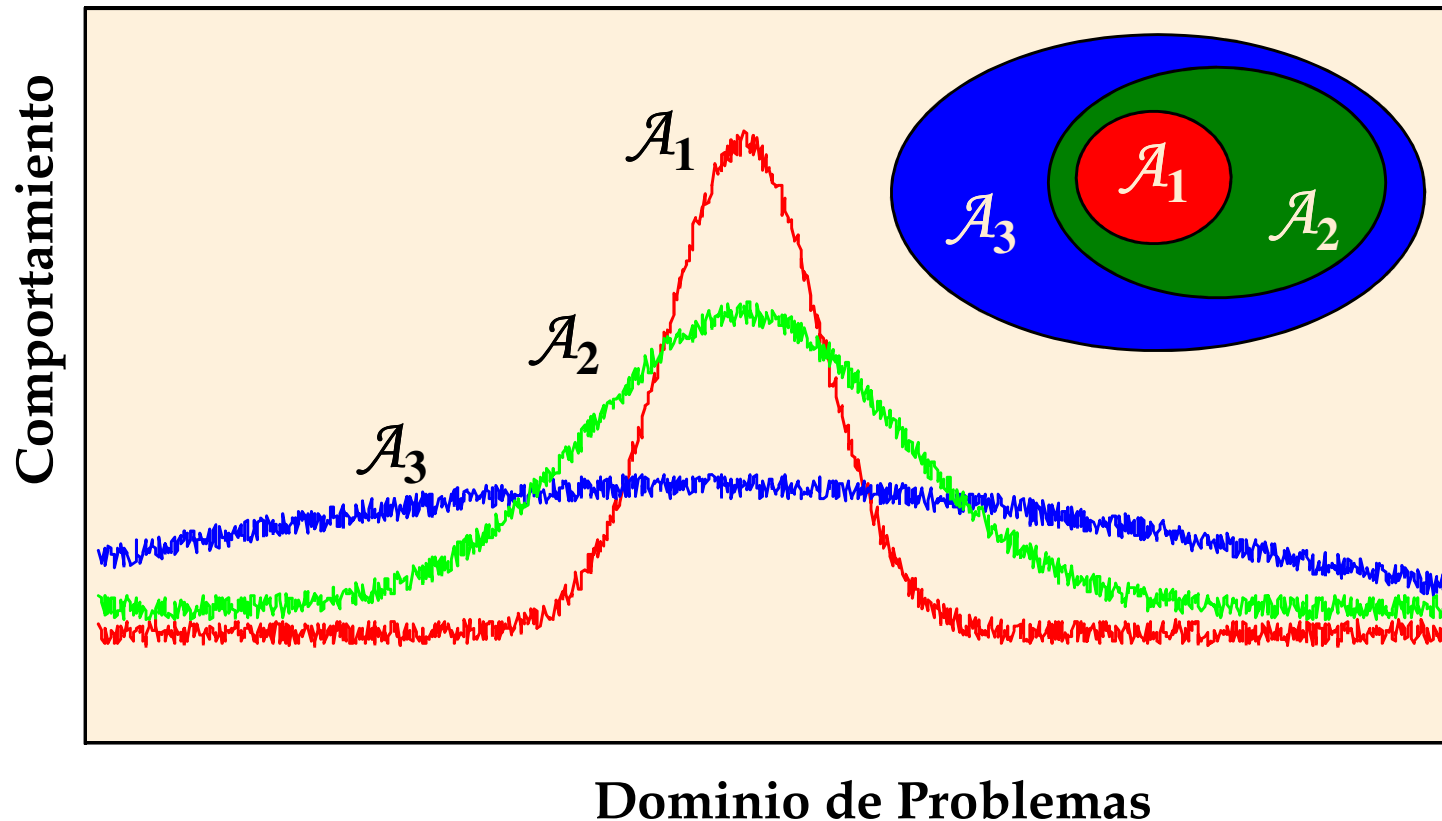
$$\sum_f E(\vec{c} / f, m, a) = \sum_f E(\vec{c} / f, m, b)$$



¿Por qué hibridar?

Los límites de los AEs

Implicaciones de NFL (II): Ganar en un cierto dominio implica perder en los restantes



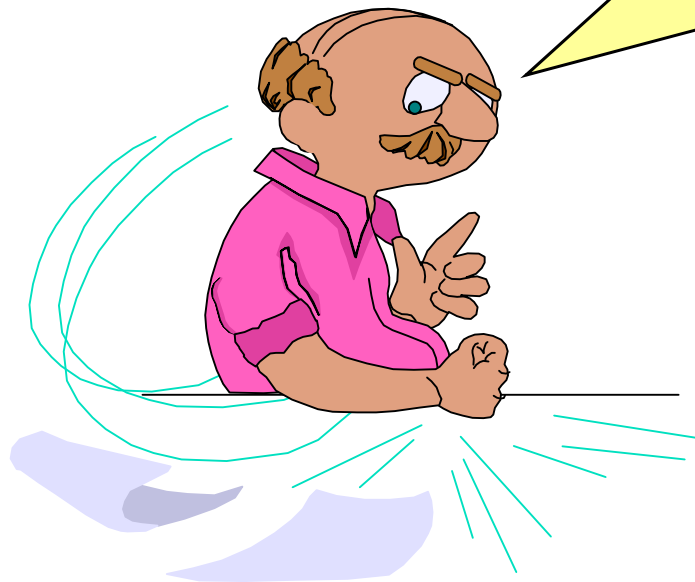
¿Por qué hibridar?

Los límites de los AEs

Discusión

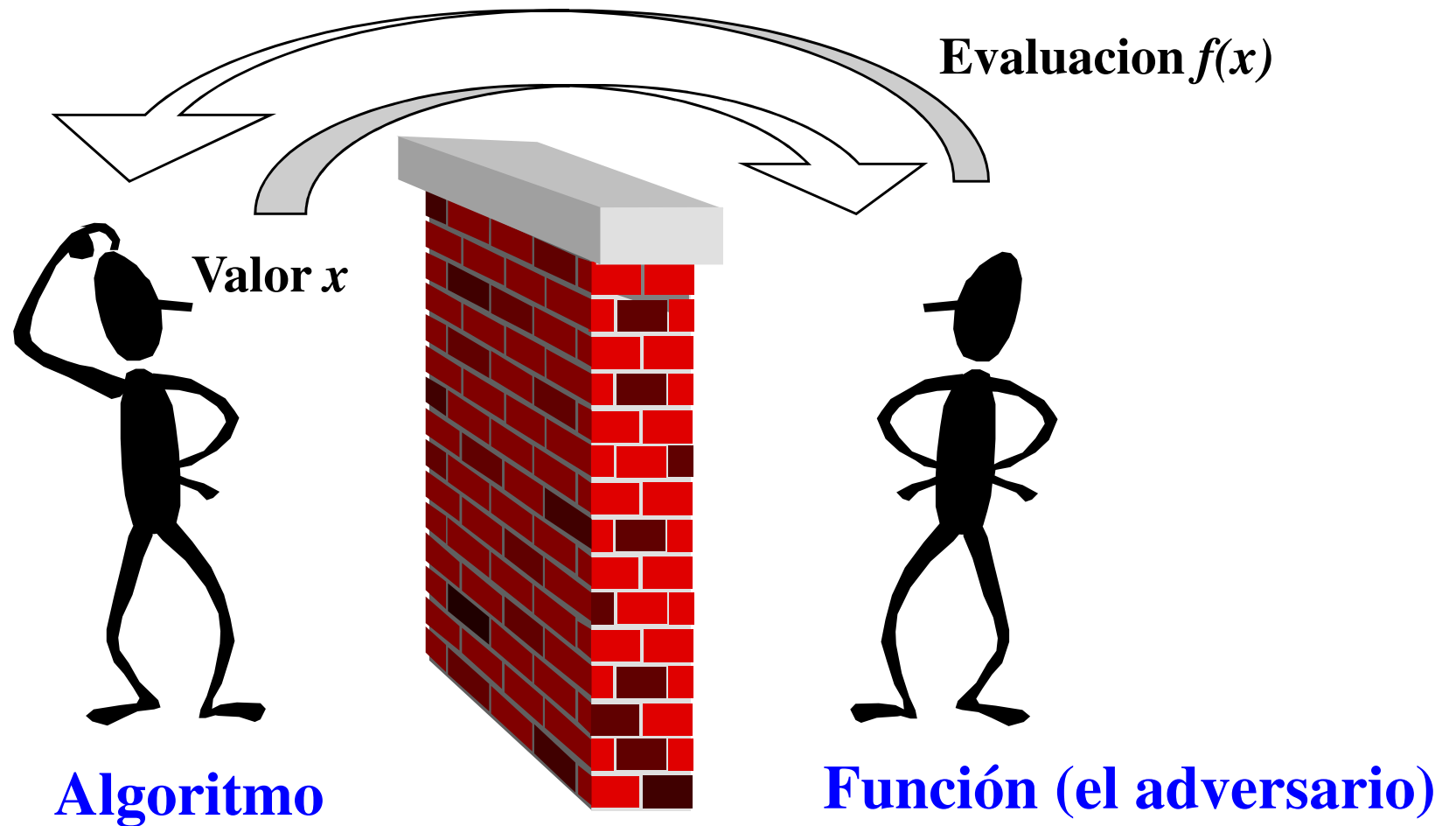
¿La búsqueda aleatoria se comporta como nuestro algoritmo?

¡Hay que disculpar algunas cuestiones en el teorema de NFL!



¿Por qué hibridar?

Los límites de los AEs



¿Por qué hibridar?

Los límites de los AEs

Consideremos los diferentes grados de conocimiento del problema:

1. Conocimiento perfecto
2. Conocimiento parcial
3. Poco conocimiento
4. Muy poco conocimiento
5. Ningún conocimiento (NFL)

Los resultados del teorema NFL son críticos tanto en cuanto (2) es comparado con (5).

¿Por qué hibridar?

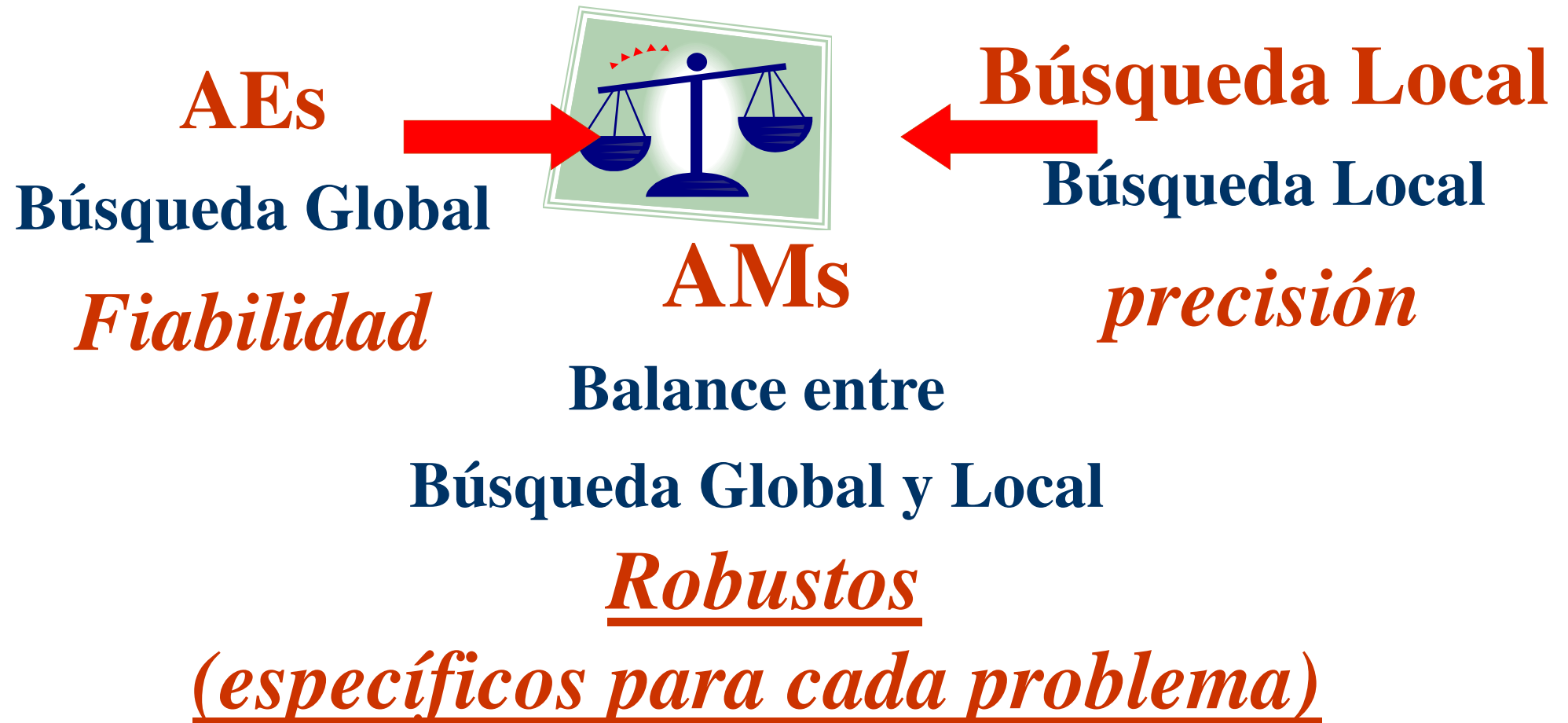
Los límites de los AEs

Los AEs pueden mejorar su comportamiento con conocimiento

- El conocimiento sobre el problema debe ser incluido en el algoritmos de búsqueda.
- Lawrence Davis resaltó esta aproximación a finales de los 80 y primeros de los 90:
 - Utilizó representación del problema **ad-hoc**.
 - **Introdujo heurísticas** específicas del problema en los AEs. Se obtiene un **AE híbrido**.
- Usualmente, las heurísticas específicas son algoritmos de búsqueda local.

¿Por qué hibridar?

Los límites de los AEs



SUMARIO ALGORITMOS MEMÉTICOS

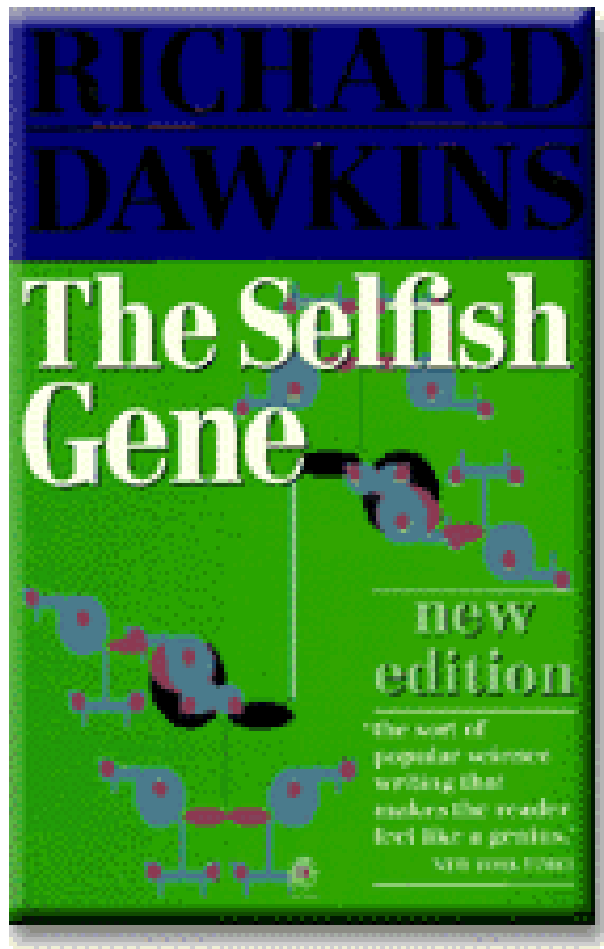
- **¿Qué es un Algoritmo Memético?**
- **¿Por qué hibridar? Los límites de los Algoritmos Evolutivos**
- **Algoritmos Meméticos: Introducción**
- **Sobre el Diseño de Algoritmos Meméticos**
- **Algoritmos Meméticos: Estudios Recientes**
- **Conclusiones**

Algoritmos Meméticos

- Los **Algoritmos Meméticos** (AMs) se construyen sobre la noción de *meme*.
- Significado: Unidad de imitación, análoga a un gen pero en el contexto de la “evolución cultural”
- El Término fue introducido por Richard Dawkins en el libro “The Selfish Gene” (“El Gen Egoísta”) (University Press, 1976)



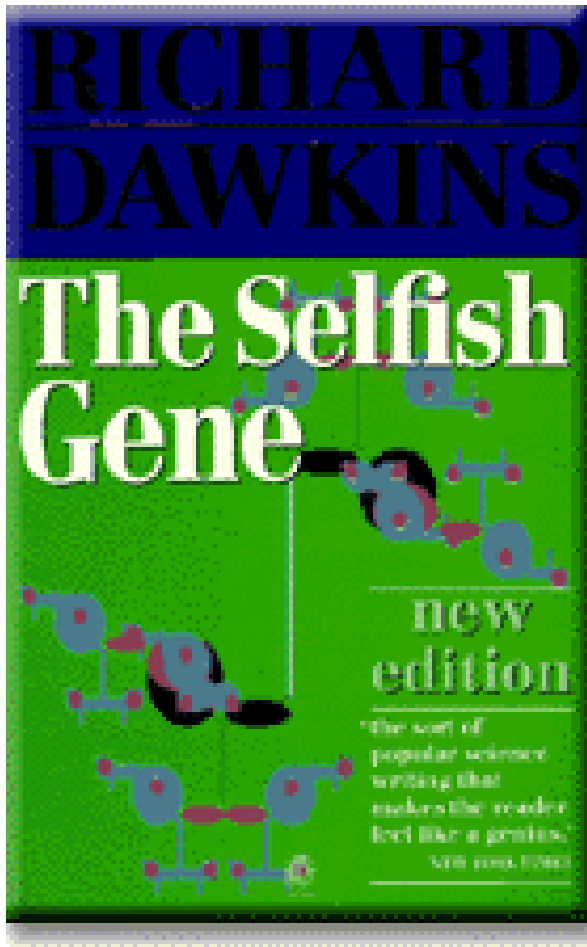
Algoritmos Meméticos



« Examples of **memes** are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches. Just as genes propagate themselves in the gene pool by leaping from body to body via sperms or eggs, so **memes** propagate themselves in the **meme pool** by leaping from brain to brain via a process which, in the broad sense, can be called imitation.»

R. Dawkins, 1976

Algoritmos Meméticos



«Ejemplos de **memes** son melodías, ideas, frases echas, modas en la vestimenta, formas de hacer vasijas, o de construir bóvedas. Del mismo modo que los genes se propagan en el acervo genético a través de gametos, los “**memes**” se propagan en el acervo memético saltando de cerebro a cerebro en un proceso que, en un amplio sentido, puede denominarse imitación.»

R. Dawkins, 1976

Algoritmos Meméticos



A **Memetic Algorithm** is a population of **agents** that alternate periods of **self-improvement** (via local search) with periods of **cooperation** (via recombination), and **competition** (via selection).

P. Moscato, 1989

Moscato, P.A. (1989). On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Caltech Concurrent Computation Program Report 826, Caltech, Pasadena, California.

Algoritmos Meméticos

TERMINOLOGÍA EN ALGORITMOS MEMÉTICOS

- En los **Algoritmos Meméticos** se utiliza el término de agentes en lugar de individuos ya que se consideran una extensión de los segundos.
- Tanto la selección como la actualización (reemplazo), son procesos puramente competitivos.
- La reproducción es la encargada de crear nuevos agentes (cooperación). Aunque puede aplicarse una gran variedad de operadores de reproducción, existen básicamente dos: Recombinación y Mutación.

Algoritmos Meméticos

TERMINOLOGÍA EN ALGORITMOS MEMÉTICOS

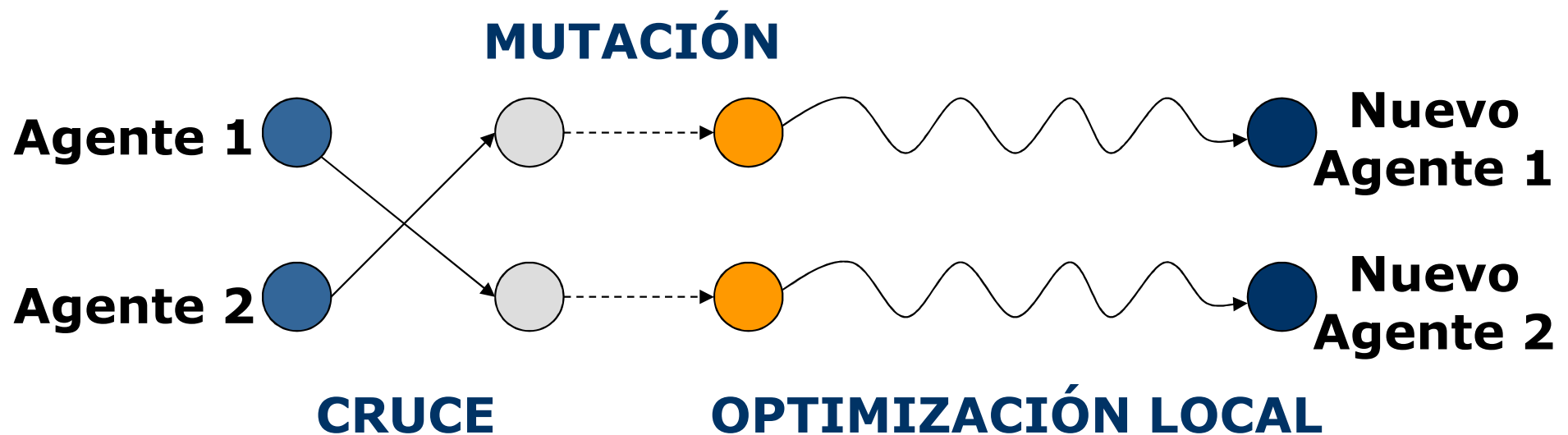
- Recombinación: Realiza el proceso de cooperación.

Crea nuevos agentes utilizando principalmente la información extraída de los agentes re combinados.

Se suele hablar de combinación inteligente de información.

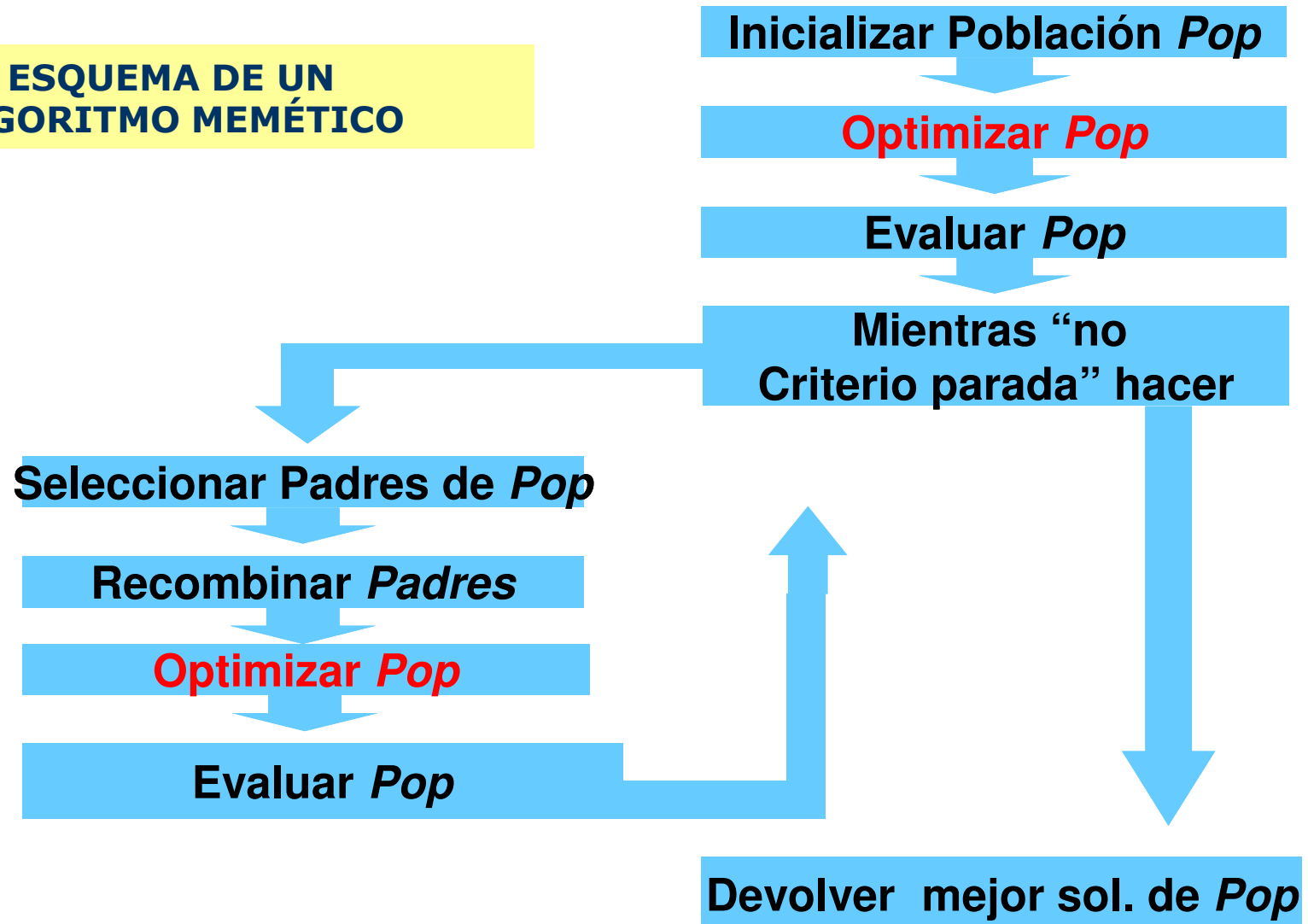
- Mutación: Permite incluir información externa creando nuevos agentes mediante modificación parcial del agente mutado.

Algoritmos Meméticos



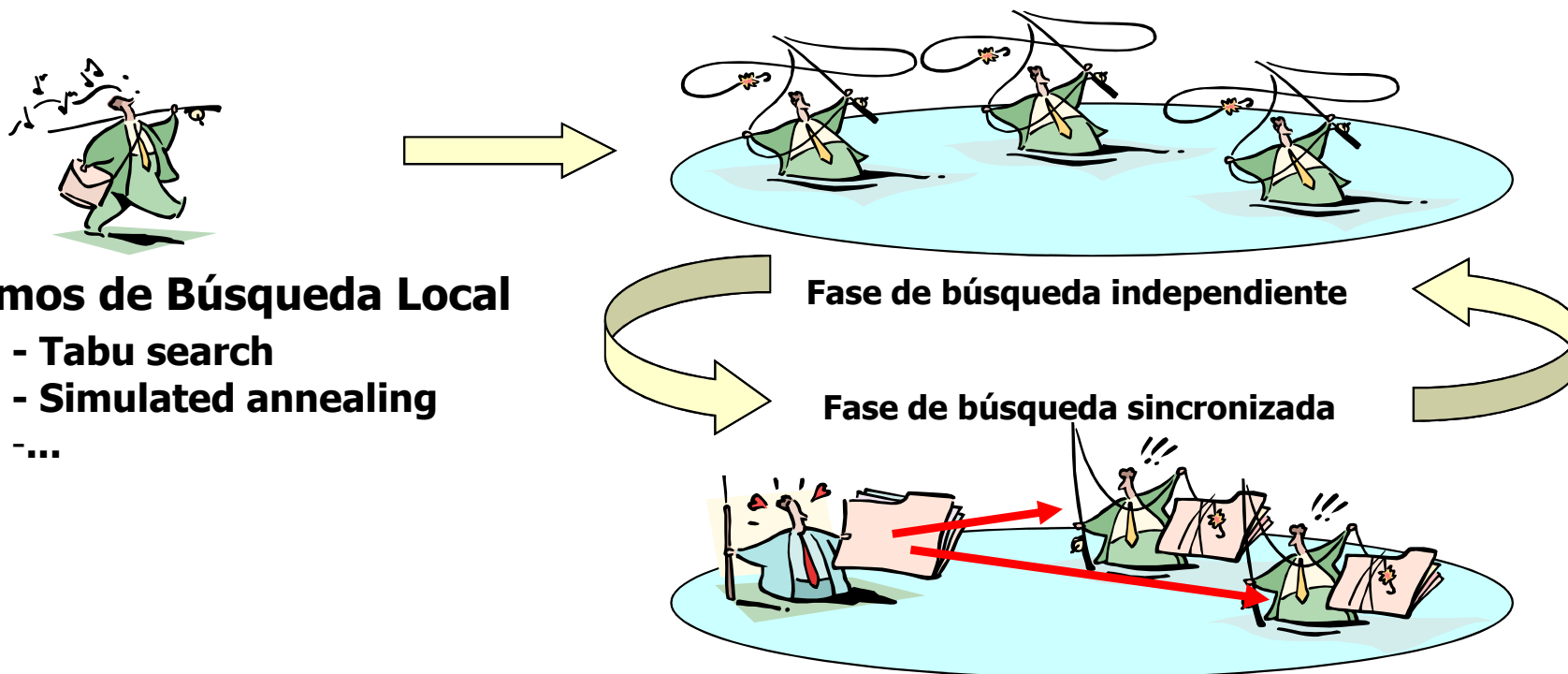
Algoritmos Meméticos

ESQUEMA DE UN ALGORITMO MEMÉTICO



Algoritmos Meméticos

OBJETIVO: ¡Ir con los mejores!



ALGORITMOS MEMÉTICOS

SUMARIO

- **¿Qué es un Algoritmo Memético?**
- **¿Por qué hibridar? Los límites de los Algoritmos Evolutivos**
- **Algoritmos Meméticos: Introducción**
- **Sobre el Diseño de Algoritmos Meméticos**
- **Algoritmos Meméticos: Estudios Recientes**
- **Conclusiones**

Sobre el diseño de Algoritmos Meméticos

- Cuando se aborda el diseño de un AM efectivo para un cierto problema, hay que partir de la base de que no existe un procedimiento sistemático para tal fin (de lo contrario entraría en conflicto con los resultados del Teorema NFL).
- Únicamente pueden considerarse heurísticas de diseño, que probablemente resultarán en un AM efectivo, pero que obviamente no lo pueden garantizar.

Sobre el diseño de Algoritmos Meméticos

- ¿Cuándo se aplica el Algoritmo de Búsqueda Local?
- ¿Sobre qué agentes se aplica?
- ¿Qué uso se hace del agente optimizado?
- ¿Cómo se aplica el optimizador local? (Intensidad de la aplicación)
- Uso de conocimiento: Otras hibridaciones
- ¿Qué Algoritmo de Búsqueda Local se utiliza?
Depende del problema a optimizar

Sobre el diseño de Algoritmos Meméticos

¿Cuándo y sobre qué agentes se aplica?

- Los optimizadores locales, considerados como un operador más, pueden aplicarse de diferentes formas:
 - En la fase de inicialización de la población
 - En cada generación o cada cierto número
 - Como fin del ciclo reproductivo o durante los operadores de recombinación
 - ...

NOTA: Para que un algoritmo híbrido sea considerado AM, la Búsqueda Local siempre debe aplicarse dentro del proceso evolutivo.

Sobre el diseño de Algoritmos Meméticos

¿Cuándo y sobre qué agentes se aplica?

- A toda la población, o
- sólo a un subconjunto de ella
 - sobre el mejor.
 - sobre representantes de clases tras un proceso de agrupación).
 - Se puede utilizar una probabilidad de actuación de la búsqueda local.
- Sobre los agentes resultantes de la reproducción o sobre los agentes de la población globalmente.
- ...

Sobre el diseño de Algoritmos Meméticos

¿Qué uso se hace del agente optimizado?

Dos modelos (clásicos)

- **Lamarckiano**

- El agente resultante del proceso de optimización local se introduce en la población (cede su genotipo) y reemplaza en la población al agente sobre el que se inició el proceso o al más cercano.

- **Baldwiniano**

- El agente inicial del proceso de optimización local recibe el fitness del agente final pero no su genotipo (cede su fitness)

Sobre el diseño de Algoritmos Meméticos

¿Cómo se aplica el optimizador local?

- Al aplicar los optimizadores locales, es esencial regular adecuadamente el equilibrio entre
 - anchura (**frecuencia de aplicación del optimizador**)
(uso de probabilidad de actuación de la Búsqueda Local: p_{LS})
 - profundidad (**intensidad del optimizador**)
 - AMs baja intensidad (pocas iteraciones del optimizador local/alta frecuencia)
 - AMs alta intensidad (muchas iteraciones del optimizador local/baja frecuencia)

SUMARIO ALGORITMOS MEMÉTICOS

- **¿Qué es un Algoritmo Memético?**
- **¿Por qué hibridar? Los límites de los Algoritmos Evolutivos**
- **Algoritmos Meméticos: Introducción**
- **Sobre el Diseño de Algoritmos Meméticos**
- **Algoritmos Meméticos: Estudios Recientes**
- **Conclusiones**

Algoritmos Meméticos: Estudios Recientes

N. Krasnogor and J.E. Smith.

A tutorial for competent memetic algorithms: model, taxonomy and design issues.
IEEE Transactions on Evolutionary Computation 9(5):474- 488, 2005.

Y.S. Ong and M.-H. Lim and N. Zhu and K.W. Wong.

Classification of Adaptive Memetic Algorithms: a Comparative Study
IEEE Transactions on System, Man. and Cybernetics. Part B: Cybernetics 36:1, 141-152, 2006.

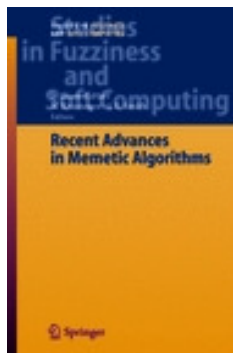
J. E. Smith. Coevolving Memetic Algorithms: A Review and Progress Report. IEEE Transactions on System, Man, and Cybernetics. Part B: Cybernetics 37:1, 2007, 6-17.

Y.S. Ong, N. Krasnogor, H. Ishibuchi (Eds.)

SPECIAL ISSUE ON MEMETIC ALGORITHMS.

**IEEE TRANSACTIONS ON
SYSTEMS, MAN, AND
CYBERNETICS
PART B: CYBERNETICS**

IEEE Transactions on System, Man. and Cybernetics. Part B: Cybernetics Vol. 37, No. 1, Feb 2007



Recent Advances in Memetic Algorithms
Studies in Fuzziness and Soft Computing, Vol. 166
Hart, William E.; Krasnogor, N.; Smith, J.E. (Eds.)
2005, X, 408 p., Hardcover
ISBN: 3-540-22904-3

Algoritmos Meméticos: Estudios Recientes

Algoritmos Meméticos Multiobjetivo

■ **M-PAES**

M-PAES: a memetic algorithm for multiobjective optimization

Knowles, J.D.; Corne, D.W.;

**Evolutionary Computation, 2000. Proceedings of the 2000 Congress on
Volume 1, 16-19 July 2000 Page(s):325 - 332 vol.1**

■ **MOGLS**

Genetic Local Search for Multi-Objective Combinatorial Optimization

Andrzej Jaskiewicz

European Journal of Operational Research 137, 2002, 50-71.

**Balance between genetic search and local search in memetic algorithms for multiobjective
permutation flowshop scheduling**

Ishibuchi, H.; Yoshida, T.; Murata, T.;

Evolutionary Computation, IEEE Transactions on 7:2 (2003), 204 – 223

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

Meta-Lamarckian Learning (Ong et al., 2004)

- El AM que utiliza un conjunto de métodos de optimización local
- Aprende la efectividad de cada optimizador local a lo largo del proceso de evolución
- La mejor Búsqueda Local tiene grandes posibilidades de ser aplicada

Meta-Lamarckian learning in memetic algorithms

Yew Soon Ong; Keane, A.J.;

Evolutionary Computation, IEEE Transactions on, 8:2, April 2004, 99 – 110

Algoritmos Meméticos: Estudios Recientes

Ejemplo: Adaptación de la Búsqueda Local

- ***Adaptive p_{LS} mechanism*** (Lozano et al., 2004)

Real Coded Memetic Algorithms with Crossover Hill-Climbing.
M. Lozano, F. Herrera, N. Krasnogor and D. Molina.
Evolutionary Computation Vol 12(2) 273-302, 2004.

Uso de una probabilidad de actuación de la Búsqueda Local: p_{LS}

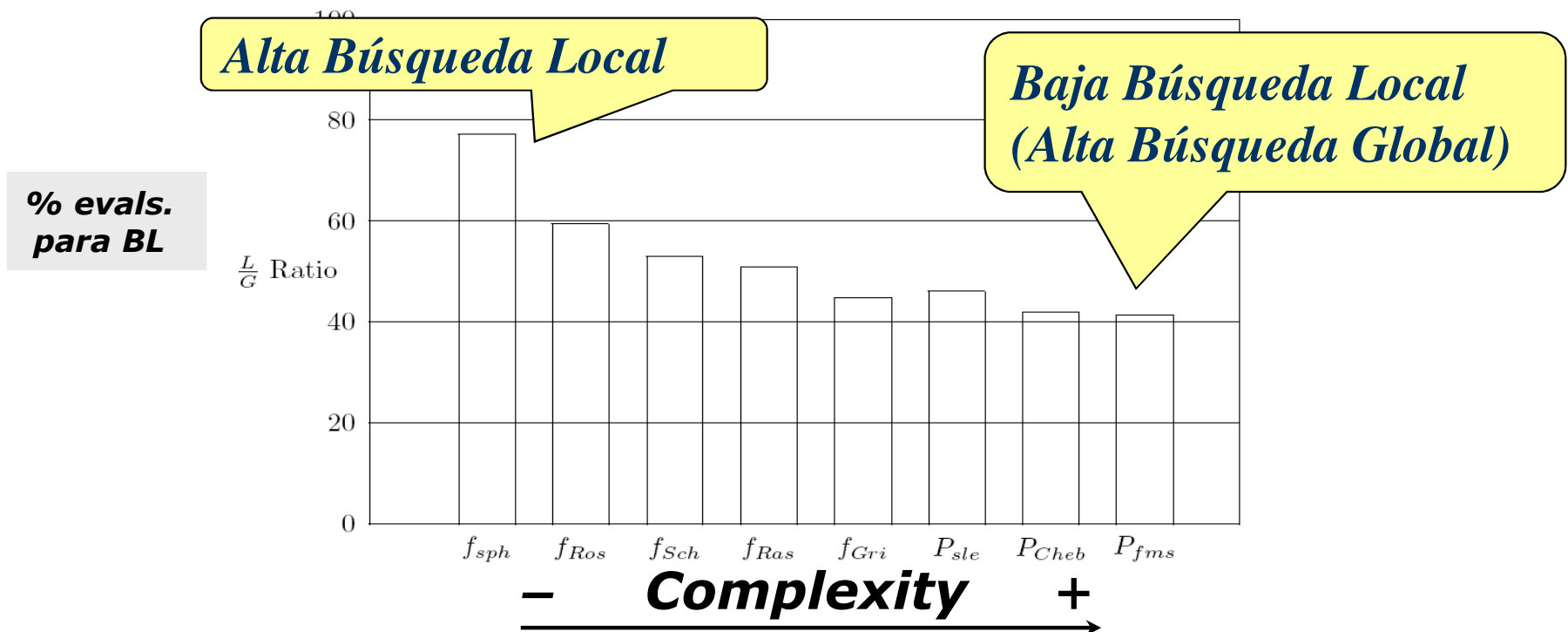
$$p_{LS} = \begin{cases} 1 & \text{if } f(c_{new}) \text{ is better than } f(c_{worst}) \\ 0.0625 & \text{Otherwise} \end{cases}$$

En muchos casos, $p_{LS} \approx 0.05$ proporciona una rápida convergencia hacia buenas soluciones (Hart, 1994)

Algoritmos Meméticos: Estudios Recientes

Ejemplo: Adaptación de la Búsqueda Local

Resultado: Ajuste del balance entre búsqueda global y local para producir un operador robusto en problemas con diferente complejidad.



Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63.

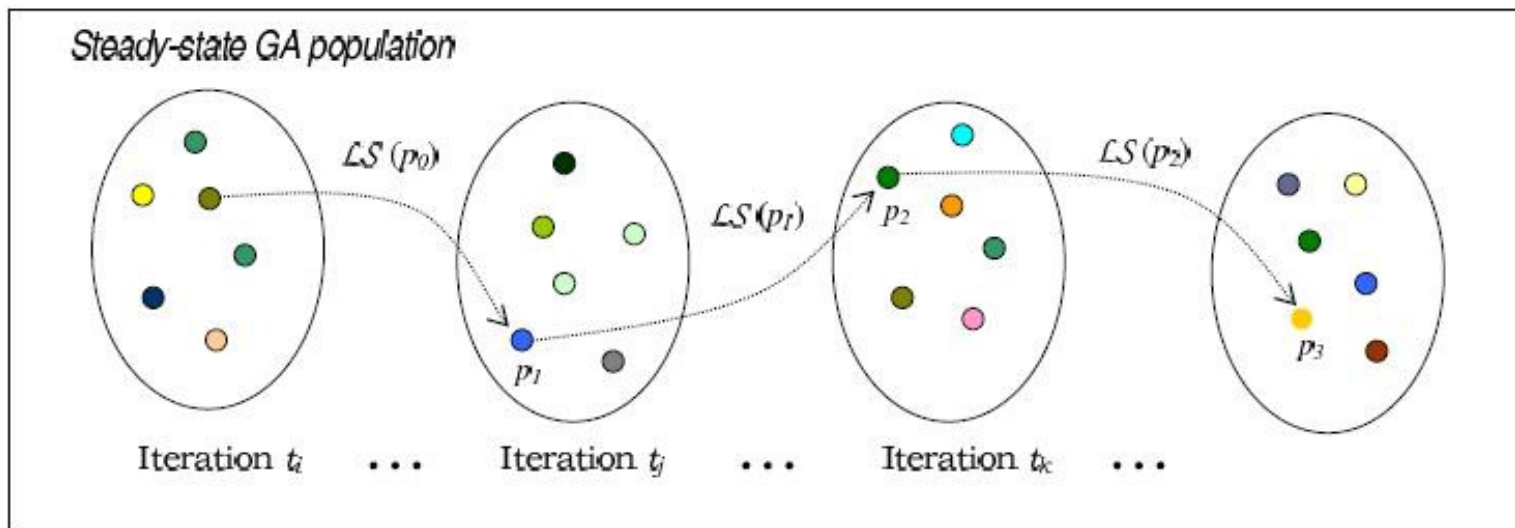


Figure 3: Example of LS chain. p_{i+1} is the final parameter value reached by the LS algorithm when it started with a value of p_i . p_0 is the default value for the strategy parameter

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

Every time the LS algorithm is applied to refine a particular chromosome, a fixed LS intensity should be considered for it, which will be called *LS intensity stretch* (I_{str}). In this way, a LS chain formed throughout n_{app} LS applications and started from solution s_0 will return the same solution as the application of the continuous LS algorithm to s_0 employing $n_{app} \cdot I_{str}$ fitness function evaluations.

After the LS operation, the parameters that define the current state of the LS processing are stored along with the reached final individual (in the steady-state GA population). When this individual is latter selected to be improved, the initial values for the parameters of the LS algorithm will be directly available. For example, if we employ the Solis and Wets' algorithm as LS algorithm, the stored strategy parameter may be the current value of the p parameter. For the more elaborate CMA-ES, the state of the LS operation may be defined by the covariance matrix (C), the mean of the distribution ($\sim m$), the size (σ), and some additional variables used to guide the adaptation of these parameters.

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

1. Generate the initial population.
2. Perform the steady-state GA throughout n_{frecc} evaluations.
3. Build the set S_{LS} with those individuals that potentially may be refined by LS.
4. Pick the best individual in S_{LS} (Let's c_{LS} to be this individual).
5. if c_{LS} belongs to an existing LS chain then
6. Initialise the LS operator with the LS state stored together with c_{LS} .
7. else
8. Initialise the LS operator with the default LS state.
9. Apply the LS algorithm to c_{LS} with an LS intensity of I_{str} (Let's c_{LS}^r to be the resulting individual).
10. Replace c_{LS} by c_{LS}^r in the steady-state GA population.
11. Store the final LS state along with c_{LS}^r .
12. If (not termination-condition) go to step 2.

Figure 4: Pseudocode algorithm for the proposed MACO model

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

MA-LSCh-CMA

Steady-state GA.

BLX- α .

Negative Assortative Mating.

BGA Mutation Operator.

Standard replacement strategy

CMA-ES as Continuous LS algorithm.



Figure 5: BLX- α

Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 9(2): 159–195.

Parameter setting. For the experiments, MA-LSCh-CMA applies BLX- α with $\alpha = 0.5$. The population size is 60 individuals and the probability of updating a chromosome by mutation is 0.125. The n_{ass} parameter associated with the negative assortative mating is set to 3. The value of the L G ratio, r_L/G , was set to 0.5, which represents an equilibrated choice. Finally, a value of $1e-8$ was assigned to the δ_{min} LS threshold.

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

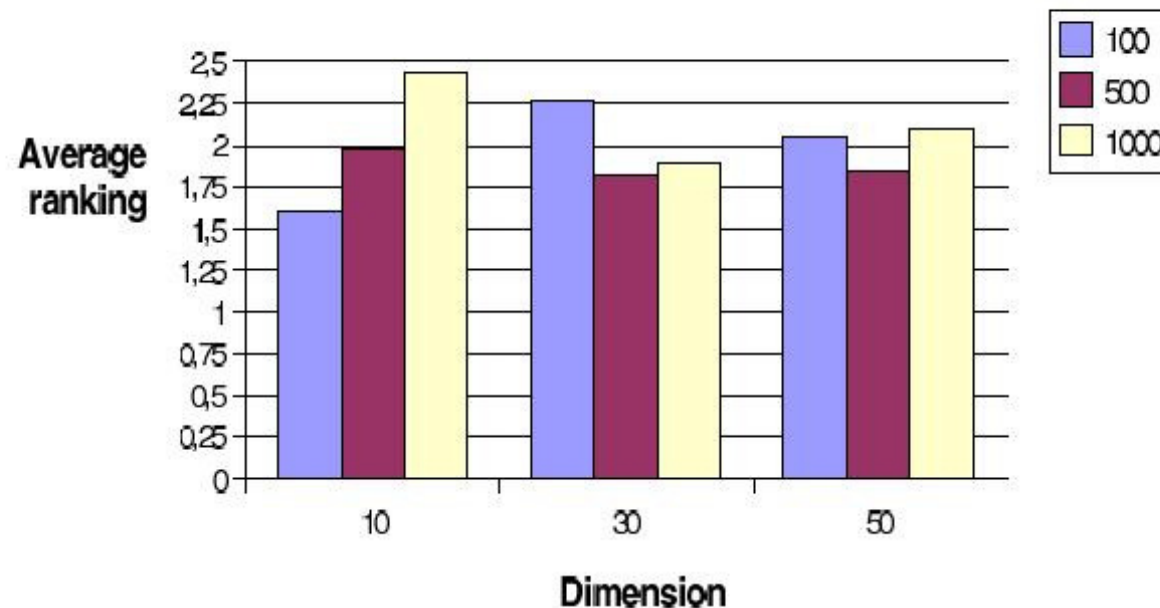


Figure 6: Rankings obtained by MA-LSCh-CMA instances with different I_{str} values

$I_{str} = 500$ is the best choice

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

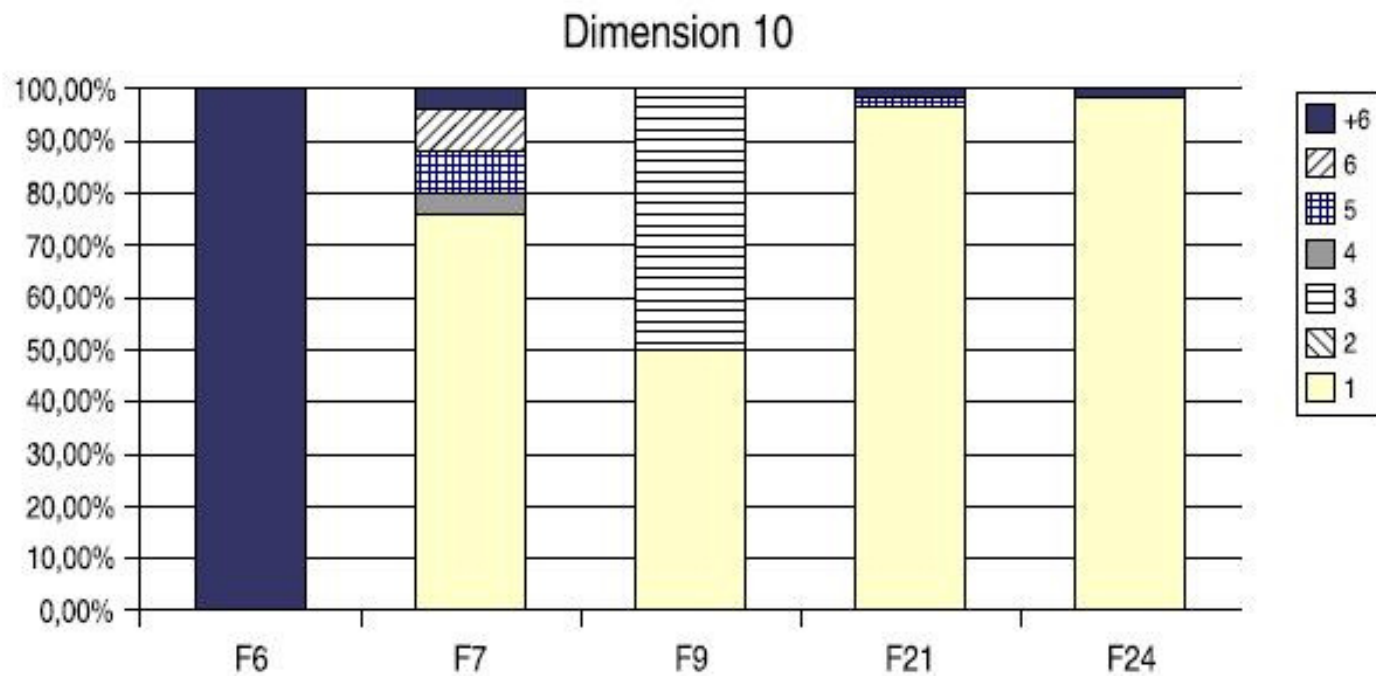


Figure 7: Percentages of LS chains with different lengths ($D = 10$)

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

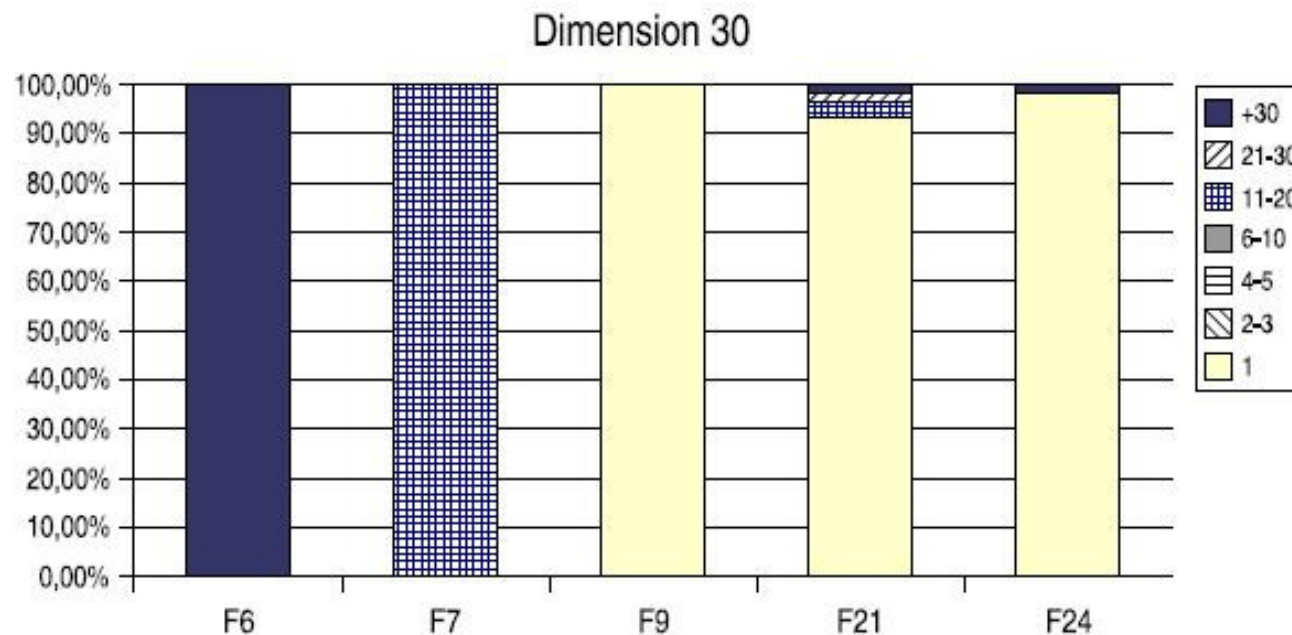


Figure 8: Percentages of LS chains with different lengths ($D = 30$)

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

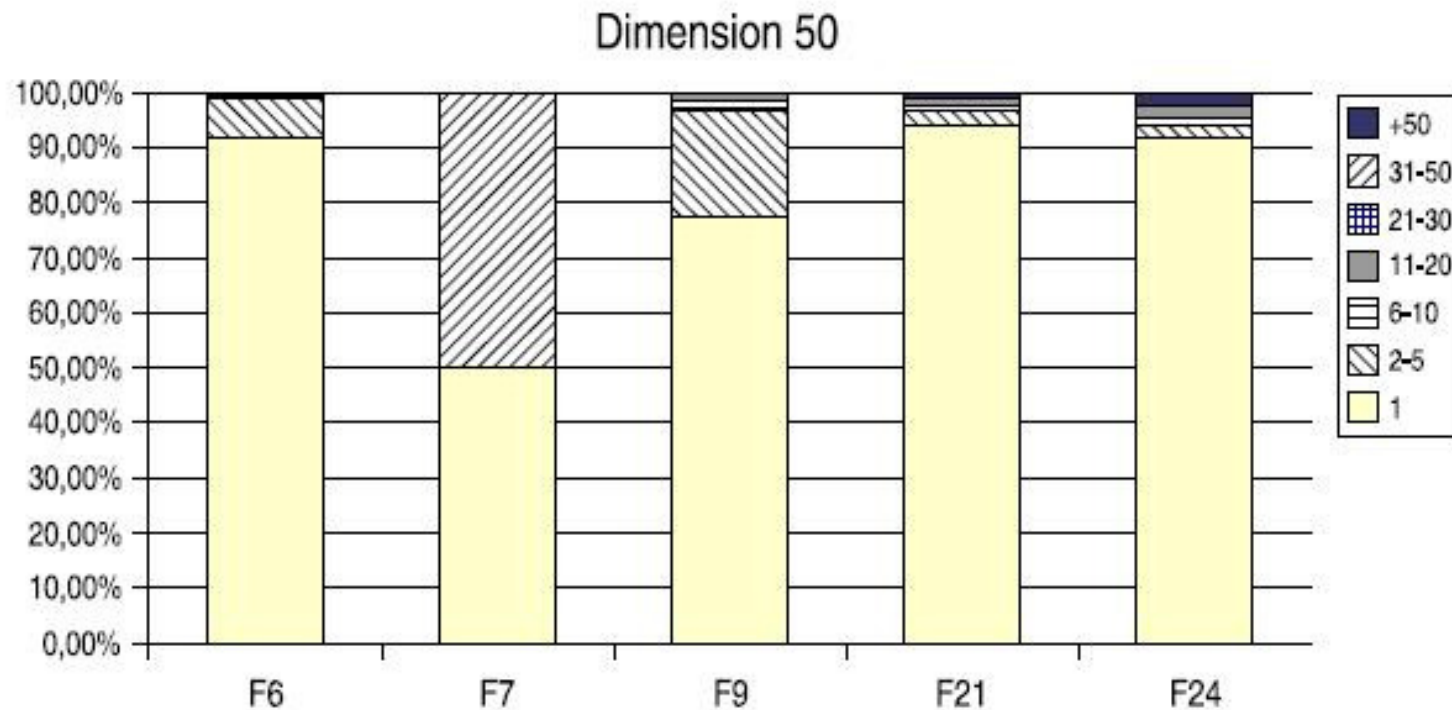


Figure 9: Percentages of LS chains with different lengths ($D = 50$)

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

Comparison with State-of-the-Art MACOs

D	$R+$ (MA-LSCh-CMA)	$R-$ (DEahcSPX)	Critical value	Sig. differences?
10	135	75	52	No
30	169.5	40.5	52	Yes
50	176.5	33.5	52	Yes

Table 7: DEahcSPX versus MA-LSCh-CMA (Wilcoxon's test with p -value = 0.05)

Noman, N. and Iba, H. (2008). Accelerating differential evolution using an adaptive local search. *IEEE Transactions on Evolutionary Computation*. 12:1 (2008)107–125.

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

Comparison with the Winner of the CEC2005

D	$R+$ (MA-LSCh-CMA)	$R-$ (G-CMA-ES)	Critical value ($p=0.05/p=0.1$)	Sig. dif? ($p=0.05$)	Sig. dif? ($p=0.1$)
10	32.5	177.5	52/60	Yes	Yes
30	139	71	52/60	No	No
50	154	56	52/60	No	Yes

Table 8: G-CMA-ES versus MA-LSCh-CMA (Wilcoxon's test with p -value = 0.05 and p -value=0.1)

Auger, A. and Hansen, N. (2005a). A restart CMA evolution strategy with increasing population size. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pages 1769-1776.

S. García, D. Molina, M. Lozano, F. Herrera, A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, [doi: 10.1007/s10732-008-9080-4](https://doi.org/10.1007/s10732-008-9080-4), 15 (2009) 617-644

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

Comparison with the Other CEC2005

Algorithm	$R+$ (MA-LSCh-CMA)	$R-$ (CEC2005)	Critical value	Sig. differences?
BLX-GL50	92.5	117.5	52	No
BLX-MA	79	131	52	No
CoEvo	157	53	52	No
DE	122	88	52	No
DMS-L-PSO	54.5	155.5	52	No
EDA	98	112	52	No
K-PCX	128	82	52	No
L-SaDE	48.5	161.5	52	Yes
SPC-PNX	95	115	52	No

Table 9: Comparison of MA-LSCh-CMA with CEC2005 competitors for $D = 10$ (Wilcoxon's test with p -value = 0.05)

Algoritmos Meméticos: Estudios Recientes

Adaptación de la Búsqueda Local

D. Molina, M. Lozano, C. García-Martínez, F. Herrera, **Memetic Algorithms for Continuous Optimization Based on Local Search Chains**. *Evolutionary Computation*, 18(1), 2010, 27–63

Comparison with the Other CEC2005

Algorithm	$R+$ (MA-LSCh-CMA)	$R-$ (CEC2005)	Critical value	Sig. differences?
BLX-GL50	166	44.5	52	Yes
BLX-MA	198	11.5	52	Yes
CoEvo	210	0	52	Yes
DE	199.5	10.5	52	Yes
K-PCX	174	36	52	Yes
SPC-PNX	169.5	40.6	52	Yes

Table 10: Comparison of MA-LSCh-CMA with CEC2005 competitors for $D = 30$ (Wilcoxon's test with p -value = 0.05)

SUMARIO **ALGORITMOS MEMÉTICOS**

- **¿Qué es un Algoritmo Memético?**
- **¿Por qué hibridar? Los límites de los Algoritmos Evolutivos**
- **Algoritmos Meméticos: Introducción**
- **Sobre el Diseño de Algoritmos Meméticos**
- **Algoritmos Meméticos: Estudios Recientes**
- **Conclusiones**

ALGORITMOS MEMÉTICOS

Comentarios finales

- Los AMs son técnicas de optimización que **explotan el conocimiento disponible** de un problema embebido en un modelo de evolución de poblaciones.
- No son un paradigma “purista” u “ortodoxo”. **Tienen muchos grados de libertad para el usuario.**
- Cuando se aborda el diseño de un AM efectivo para un cierto problema, hay que partir de la base de que **no existe un procedimiento sistemático para tal fin.**
- Han demostrado ser **más eficaces** que los AGs para diferentes problemas.

ALGORITMOS MEMÉTICOS

Bibliografía Básica

- P. Moscato, "Memetic Algorithms: A short introduction", *New Ideas in Optimization* (pp. 219-234), Corne D., Dorigo M., Glover F., McGraw-Hill-UK, 1999
- P. Moscato, C. Cotta, "A Gentle Introduction to Memetic Algorithms", *Handbook of Metaheuristics*, F. Glover, G. Kochenberger (eds.), pp. 105-144, Kluwer Academic Publishers, Boston MA, 2003
- P. Moscato, C. Cotta, "Una Introducción a los Algoritmos Memeticos", *Inteligencia Artificial. Revista Iberoamericana de IA*, No. 19, 2003, 131-148.
- W E Hart, N Krasnogor and J E Smith. "Memetic Evolutionary Algorithms", *Recent Advances in Memetic Algorithms*, Hart, William E.; Krasnogor, N.; Smith, J.E. (Eds.) 2005, 3-27.
- N. Krasnogor and J.E. Smith.
A tutorial for competent memetic algorithms: model, taxonomy and design issues.
IEEE Transactions on Evolutionary Computation 9(5):474- 488, 2005.
- Y.S. Ong and M.-H. Lim and N. Zhu and K.W. Wong.
Classification of Adaptive Memetic Algorithms: a Comparative Study
IEEE Transactions on System, Man. and Cybernetic. Part B. 36:1, 141-152, 2006.
- J. E. Smith. Coevolving Memetic Algorithms: A Review and Progress Report. IEEE Transactions on System, Man, and Cybernetics. Part B: Cybernetics 37:1, 2007, 6-17.

METAHEURÍSTICAS

TEMA 7. METAHEURÍSTICAS HÍBRIDAS

1. ALGORITMOS MEMÉTICOS

2. BÚSQUEDA DISPERSA (SCATTER SEARCH)

Bibliografía

M. Laguna, R. Martí. **Scatter Search**. Kluwer, 2002

R. Martí, M. Laguna. **Scatter Search: Diseño Básico y Estrategias Avanzadas**.
Revista Iberoamericana de Inteligencia Artificial 19 (2003) 123-130

F. Glover, M. Laguna, R. Martí. **Chapter 1: Scatter Search and Path Relinking: Advances and Applications**. In: F. Glover, G.A. Kochenberger, (Eds.). Handbook of Metaheuristics. Kluwer Academics. (2003) 1-35.

BÚSQUEDA DISPERSA

- **Orígenes**
- **Representación Gráfica – Combinación de Soluciones**
- **Esquema – Esquema Básico**
 - **Generación Diversa**
 - **Método de Mejora**
 - **Conjunto de referencia**
 - **Generación de Subconjuntos**
 - **Combinación de Soluciones**
 - **Esquema**
- **Extensiones y Mejoras**
- **Conclusiones**

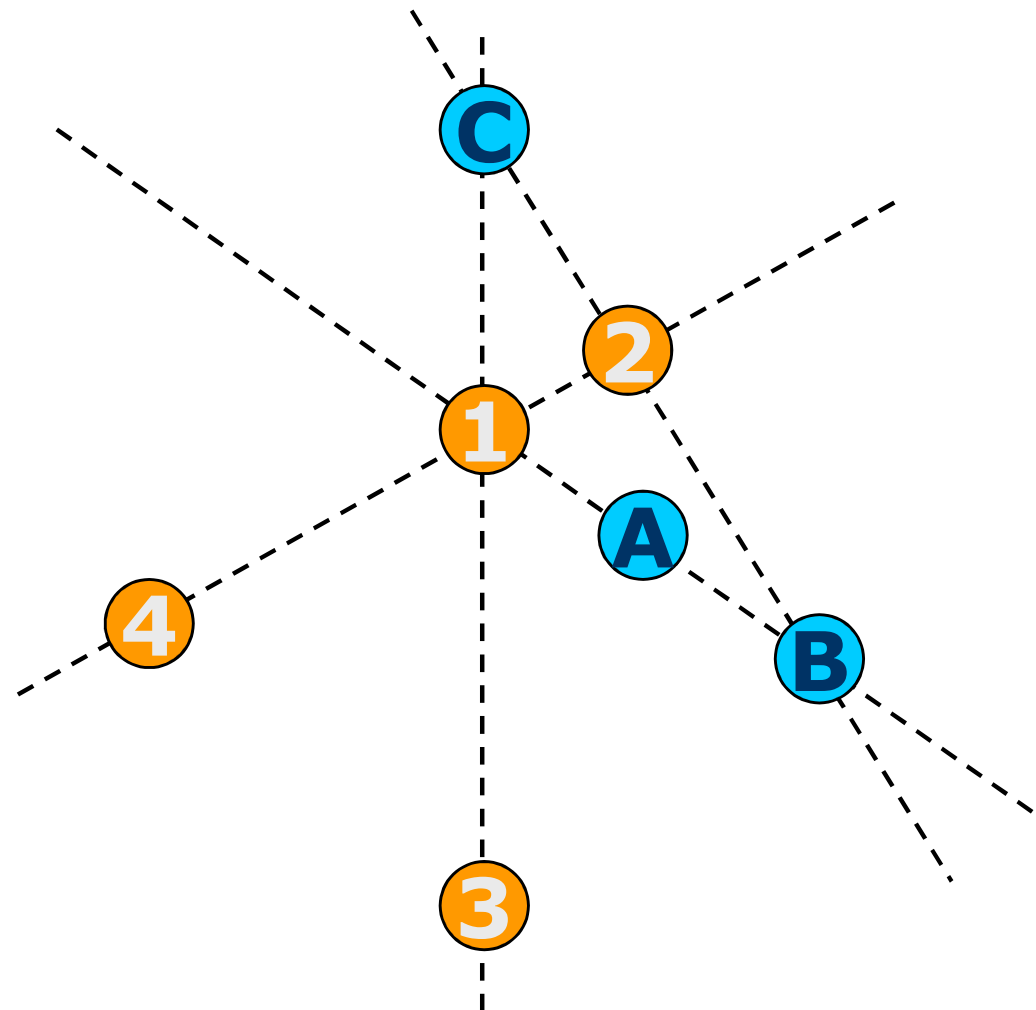
BÚSQUEDA DISPERSA

Orígenes

- Es un método evolutivo que **combina** soluciones en un conjunto de referencia para crear nuevas soluciones
- La noción de combinar elementos para crear otros nuevos se originó en la década de los 60
 - Combinar reglas de selección en *scheduling*
 - Combinar restricciones en programación entera (restricciones subrogadas)

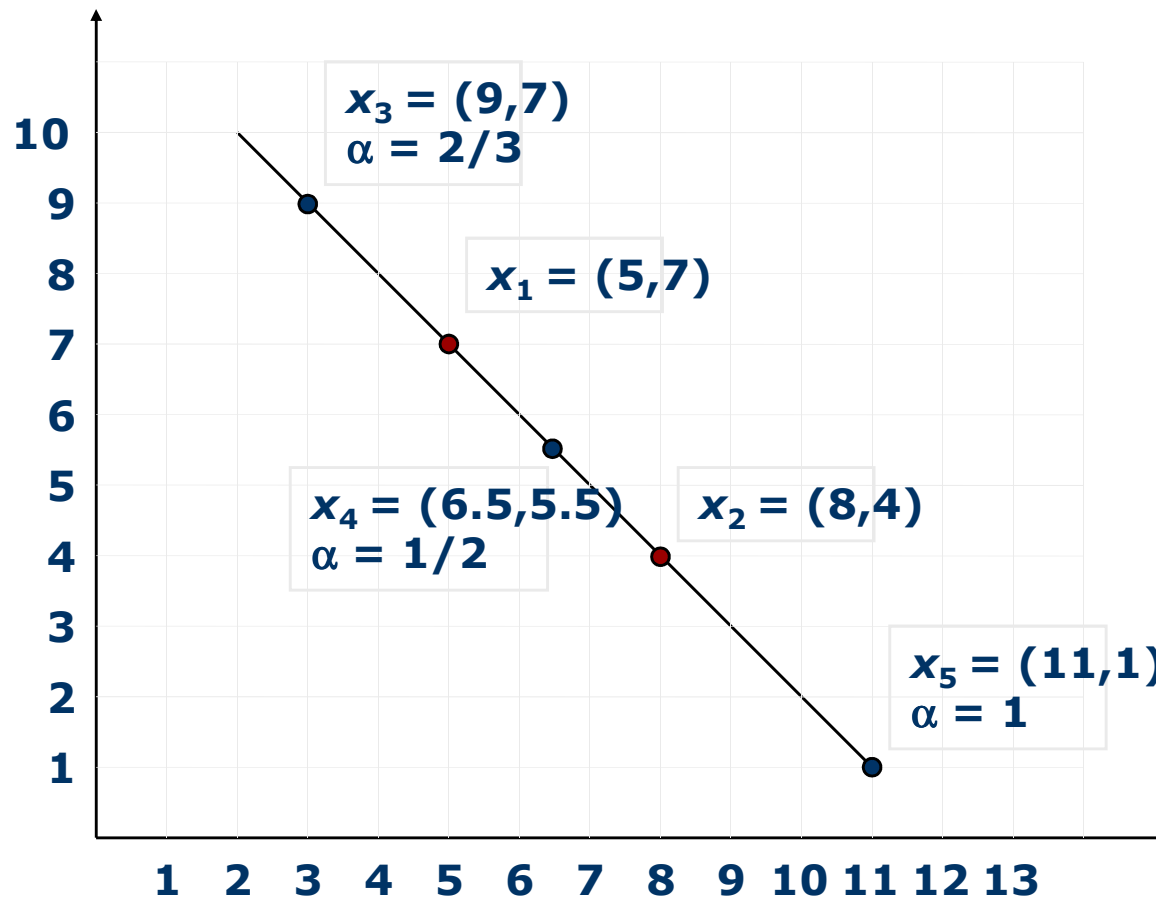
BÚSQUEDA DISPERSA

Representación Gráfica



BÚSQUEDA DISPERSA

Combinación Lineal (2D)



$$\begin{aligned}x_3 &= x_1 - \alpha \cdot (x_2 - x_1) \\x_4 &= x_1 + \alpha \cdot (x_2 - x_1) \\x_5 &= x_2 + \alpha \cdot (x_2 - x_1)\end{aligned}$$

BÚSQUEDA DISPERSA

Esquema: "Template"

1. Método de Generación Diversa †
2. Método de Mejora ^
3. Conjunto de Referencia †
 - Inicialización
 - Actualización
 - Reconstrucción
4. Generación de Subconjuntos *
5. Combinación de Soluciones ^

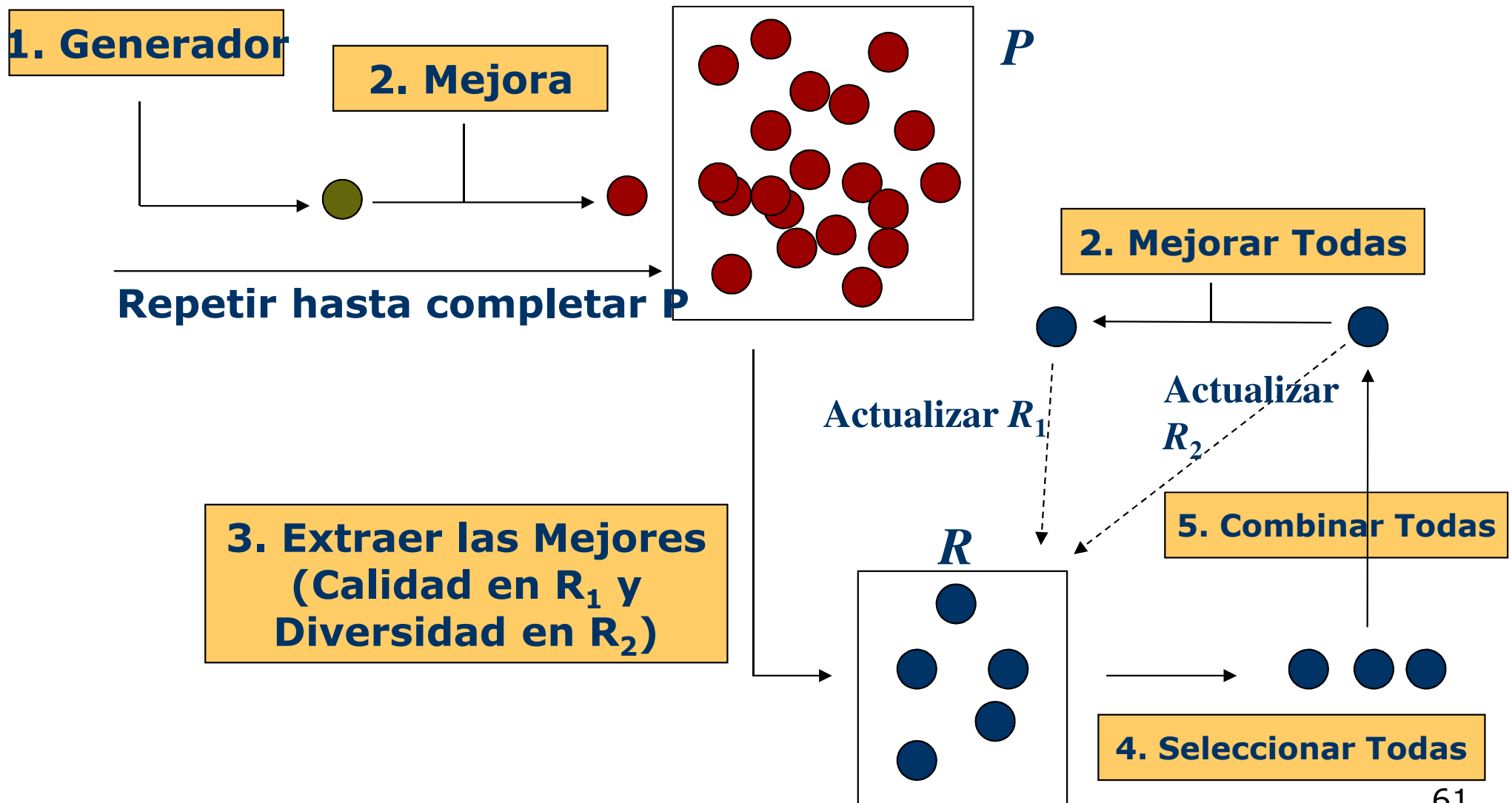
† Independiente del Problema (excepto la medida de distancia)

* Independiente del Problema

^ Dependiente del Problema

BÚSQUEDA DISPERSA

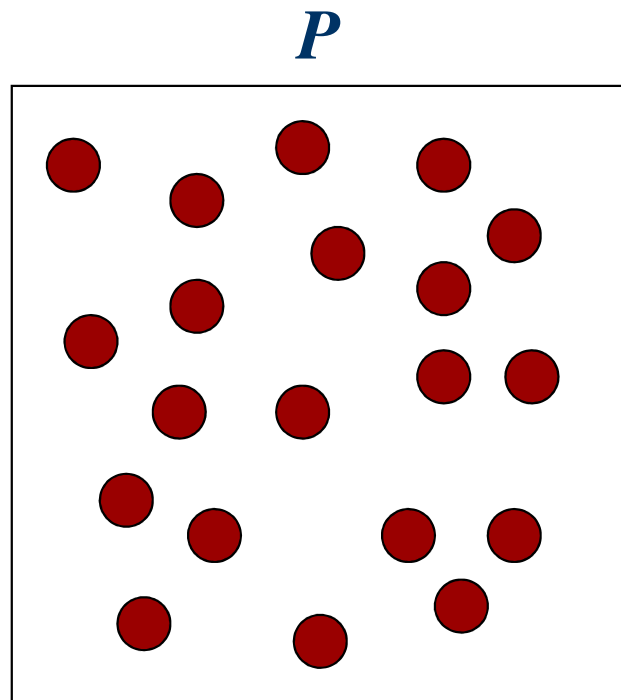
Esquema Básico



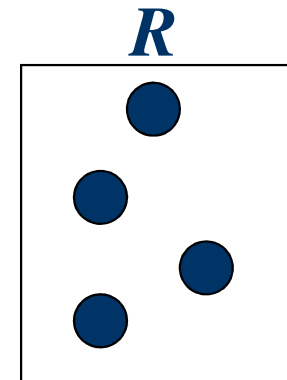
BÚSQUEDA DISPERSA

1. Generación Diversa

- Construimos un Conjunto de Soluciones P con un cierto nivel de Calidad y Diversidad, del que se extraerá el Conjunto de Referencia R , sobre el que trabaja el proceso evolutivo



$$|P| \geq 10 \cdot |R|$$

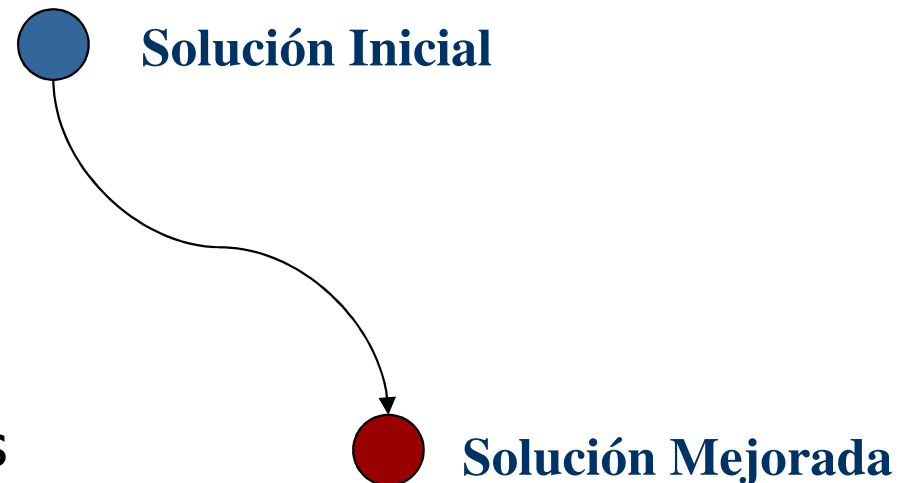


BÚSQUEDA DISPERSA

2. Método de Mejora

- Para obtener soluciones de Calidad, aplicamos un método de mejora (usualmente una “Búsqueda Local”)

- Se aplica a:
 - Soluciones Originales
 - Soluciones Combinadas

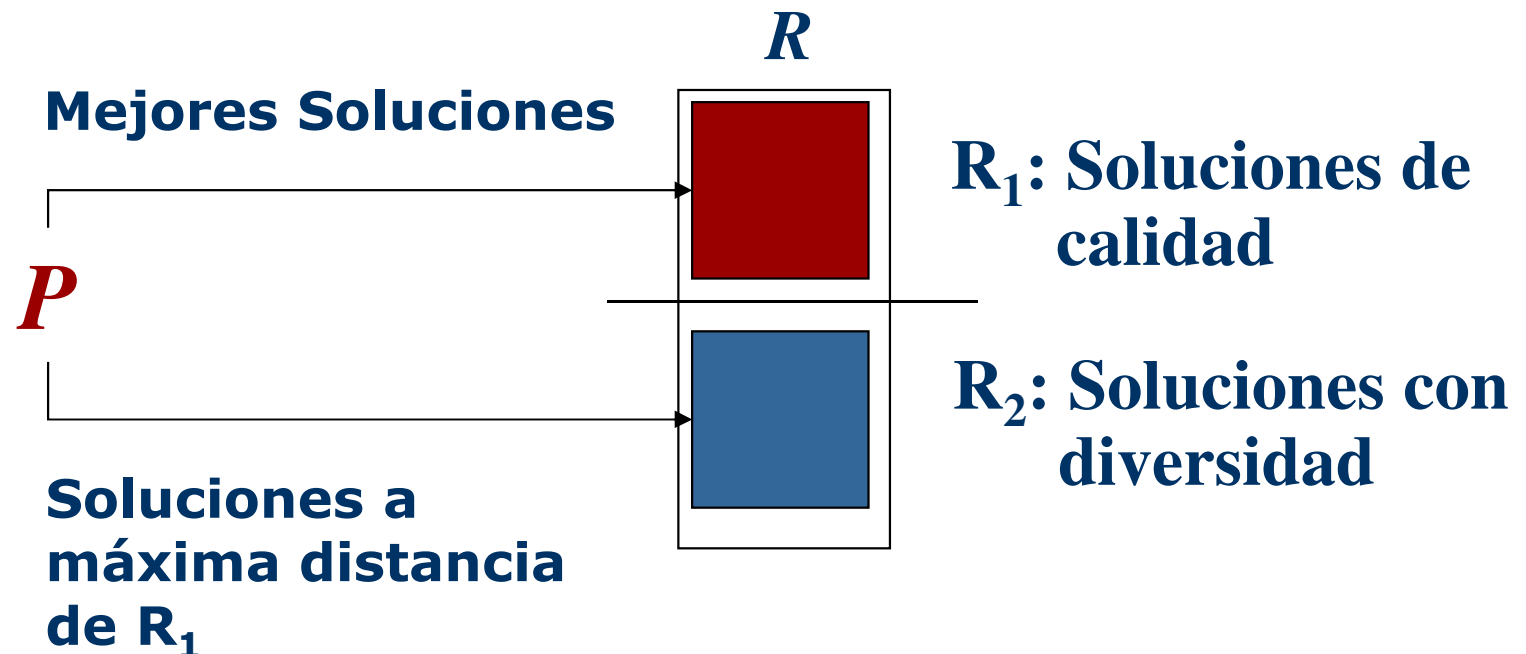


BÚSQUEDA DISPERSA

3. Conjunto de Referencia

3.1 Inicialización:

- Una parte de R se construye con soluciones de calidad y la otra por diversidad



BÚSQUEDA DISPERSA

3. Conjunto de Referencia

3.1 Inicialización:

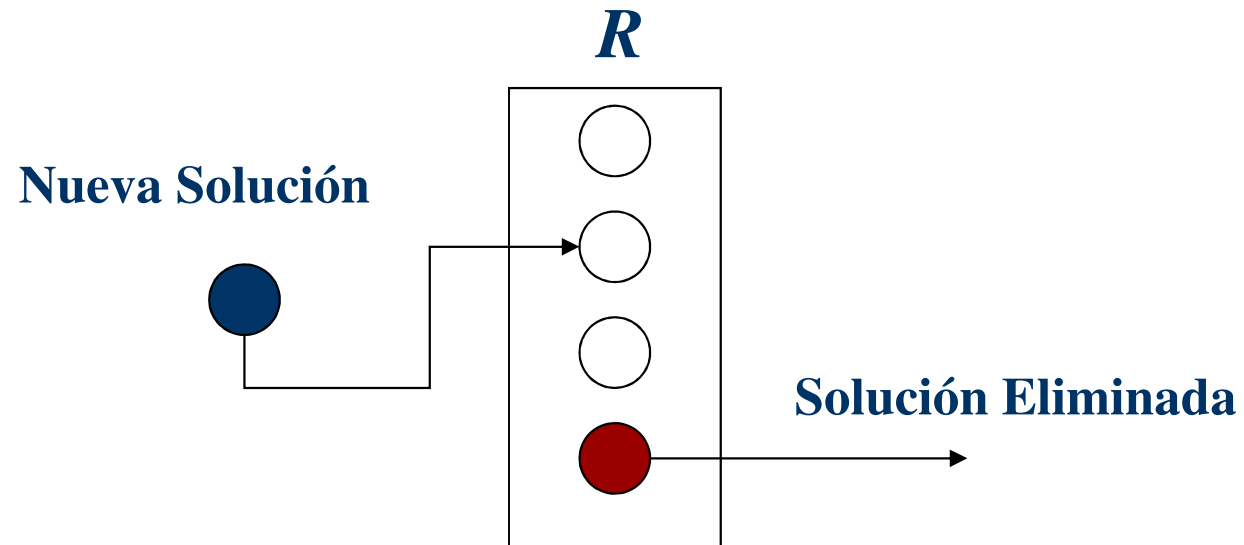
- R_1 se construye incluyendo las $|R_1|$ mejores soluciones de P
- R_2 se construye incluyendo las $|R_2|$ soluciones de P más distantes a las soluciones previamente incluidas en R_1
- Para valorar la distancia de cada solución al conjunto R_1 , se calcula la distancia a cada solución de R_1 y se obtiene la media
- La medida de distancia depende de la representación utilizada y del problema abordado

BÚSQUEDA DISPERSA

3. Conjunto de Referencia

3.2 Actualización:

- Cada solución resultado de combinar entra en el conjunto de referencia si mejora (en calidad o diversidad) a la peor
- $|R|$ se mantiene constante



BÚSQUEDA DISPERSA

3. Conjunto de Referencia

3.2 Actualización:

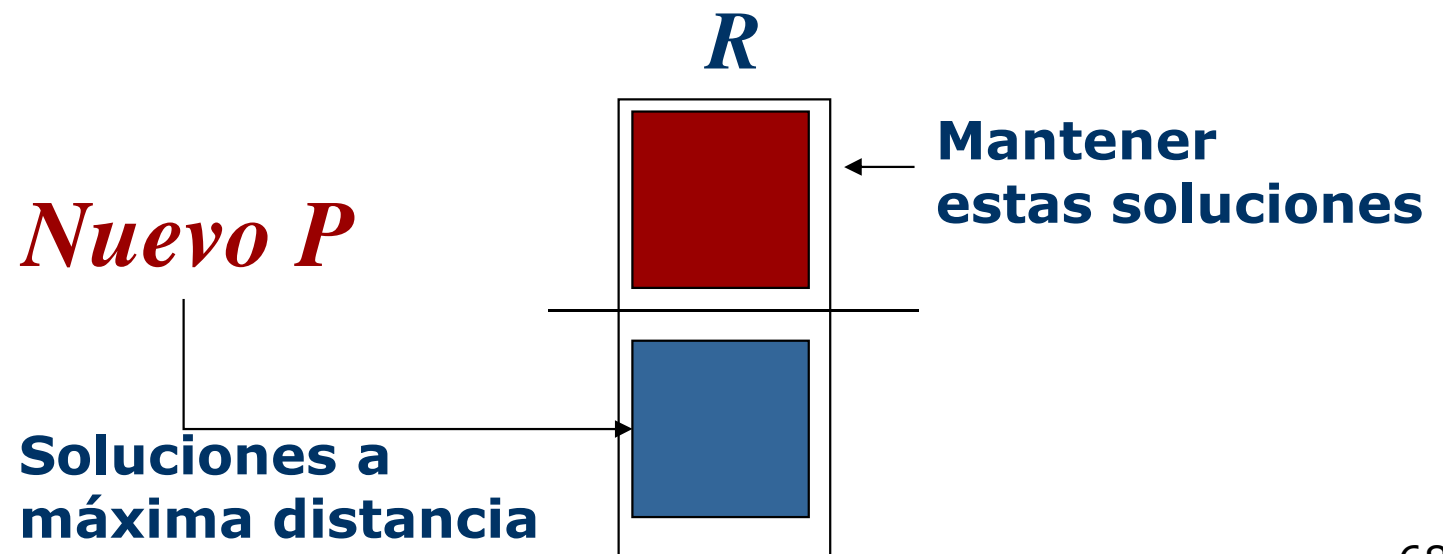
- Por ejemplo, se puede realizar el siguiente procedimiento:
 - A cada solución combinada se le aplica el método de mejora. Las soluciones optimizadas resultantes se comparan con las incluidas en el conjunto R_1 , y se actualizan aquellas que mejoran (sustituyéndose por las peores)
 - Posteriormente, se analiza la distancia de cada solución combinada (previo al método de mejora) al nuevo conjunto R creado (R_1 nuevo + R_2 antiguo). Aquellas soluciones más distantes, sustituyen a las más cercanas de R_2

BÚSQUEDA DISPERSA

3. Conjunto de Referencia

3.3 Reconstrucción

- Cuando en una iteración de la actualización no se consigue mejorar ninguna solución, se reconstruye R
- Mantenemos el subconjunto de calidad (R_1) y eliminamos el de diversidad (R_2)
- Generamos un nuevo P y extraemos $|R_2|$ soluciones diversas respecto a R_1 que pasan a formar el nuevo R_2

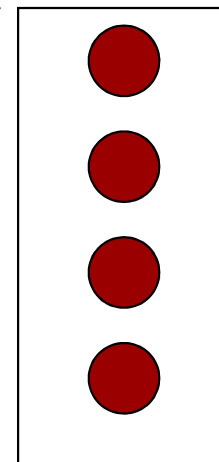


BÚSQUEDA DISPERSA

4. Generación de Subconjuntos (Selección)

- El procedimiento genera subconjuntos de R , de forma determinista, a los que se aplica el método de combinación
- Las combinaciones no se limitan a parejas de soluciones, se pueden generar ternas, cuaternas, ...
- En la práctica se ha comprobado en algunos problemas que:

**Alrededor del 80% de las soluciones de R
están generadas por combinaciones de 2 soluciones**



R

BÚSQUEDA DISPERSA

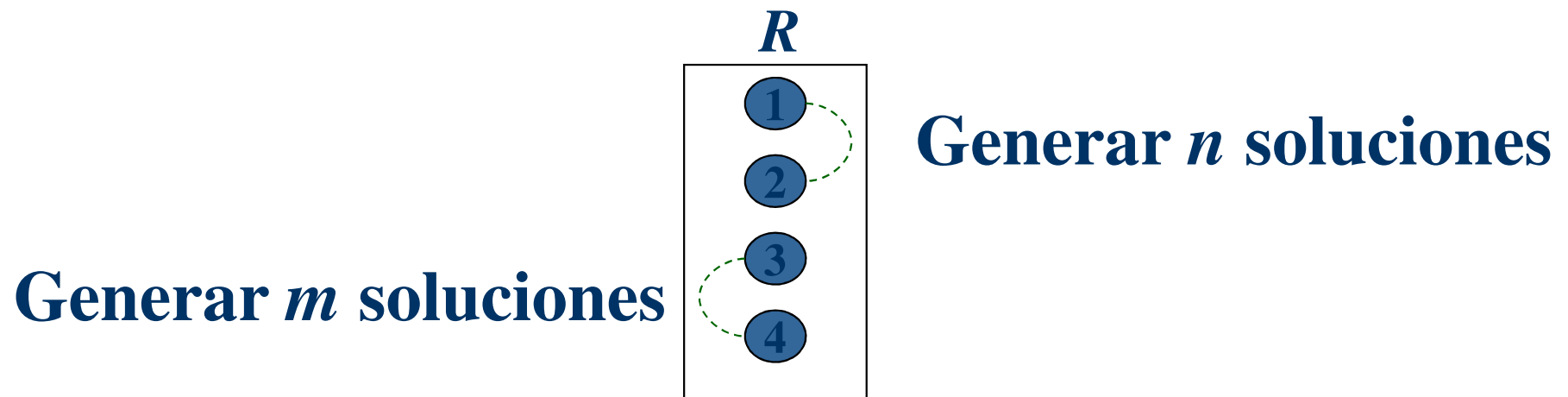
4. Generación de Subconjuntos (Selección)

- Por ejemplo, se puede seguir el siguiente procedimiento:
 - Cada subconjunto está compuesto de dos soluciones
 - Se generan todas las parejas posibles teniendo en cuenta que habrá tres grupos:
 - Combinaciones entre dos soluciones de R_1
 - Combinaciones con una solución de R_1 y otra de R_2
 - Combinaciones entre dos soluciones de R_2

BÚSQUEDA DISPERSA

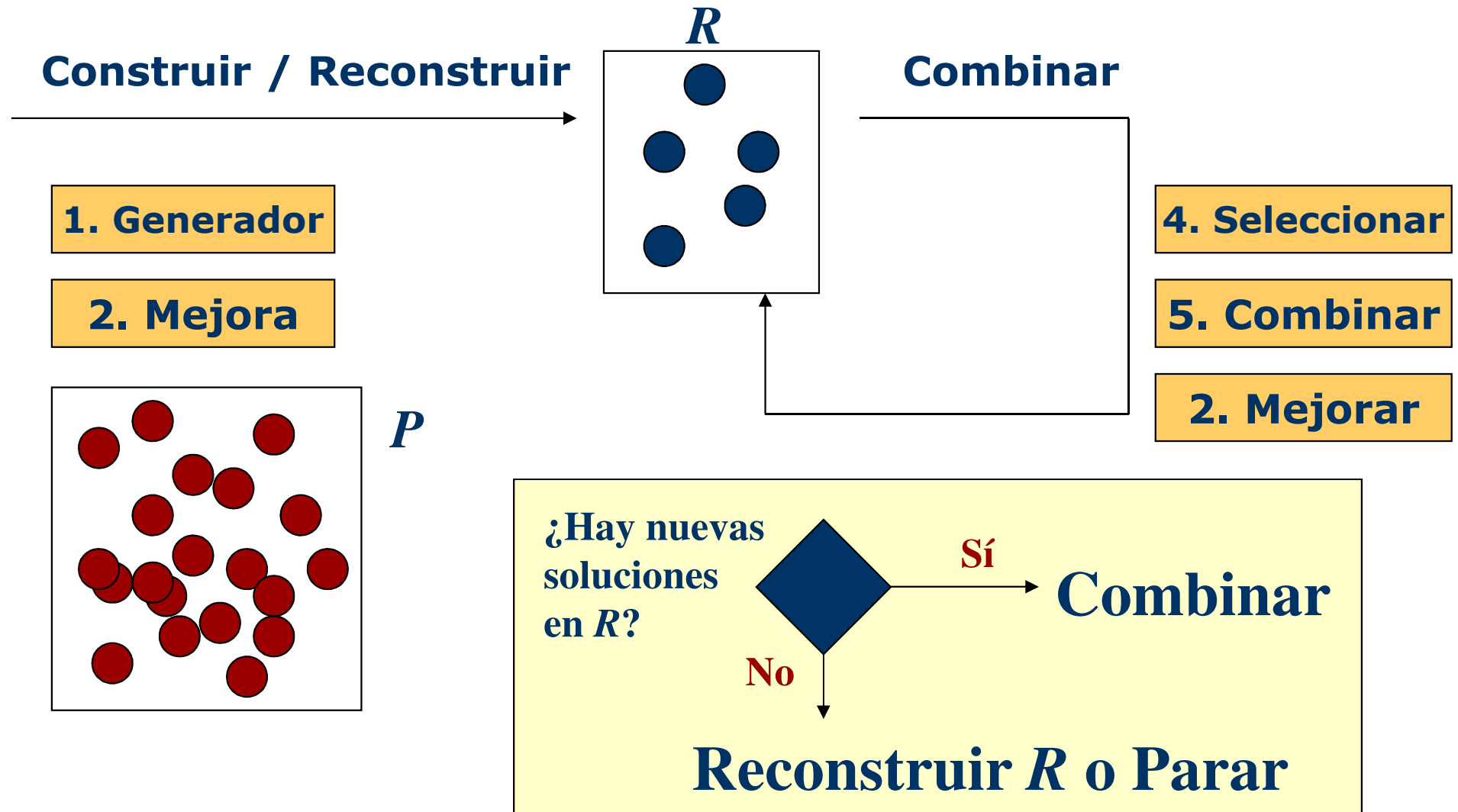
5. Combinación de Soluciones

- En la mayor parte de los problemas hay que diseñar métodos de combinación específicos
- Se pueden aplicar de forma selectiva y/o utilizar elementos aleatorios
- Muchas veces se puede emplear un operador de cruce ya existente en AG para la representación empleada
- El número de combinaciones generadas a partir de cada subconjunto se define para cada problema:



BÚSQUEDA DISPERSA

Esquema



BÚSQUEDA DISPERSA

Extensiones y Mejoras

1. Generador (Diversificación)

- Utilizar “memoria” (frecuencias) para generar de forma determinista buenas soluciones diversas

2. Método de Mejora

- Utilizar un procedimiento con “memoria” como Búsqueda Tabú en lugar de una búsqueda local
- Estudiar el tiempo que hay que dedicar a Generar y el dedicado a Mejorar

3. Actualización de R

- Estática versus Dinámica (Agresiva)
- Considerar también la Diversidad

BÚSQUEDA DISPERSA

Código de libre distribución

- <http://www.uv.es/~rmarti/sscode.html>
- Libro
 - M. Laguna R. Martí. "Scatter Search"
 - Kluwer, 2002

EJEMPLO: VIAJANTE DE COMERCIO

Método de generación diversa

- Cada vez que se quiera incluir una nueva solución en P se hará lo siguiente:
 1. En el conjunto de soluciones actualmente incluidas en P , sea $c_{i,j}$ el número de veces que la ciudad i ha estado inmediatamente delante de la ciudad j en un recorrido
 2. Generar una solución “alejada” del conjunto P actual:
 - Escoger una ciudad de inicio aleatoriamente
 - En cada paso, a partir de la ciudad actual i , escoger la próxima ciudad a visitar según la siguiente probabilidad:

$$P_{i,j} = \frac{1 - p_{i,j}}{|C_i| - 1}, \text{ donde } C_i = \{\text{ciudades aún no visitadas}\}, j \in C_i \text{ y } p_{i,j} = \frac{c_{i,j}}{\sum_{k \in C_i} c_{i,k}}$$

EJEMPLO: VIAJANTE DE COMERCIO

Método de generación diversa

(continuación)

3. Optimizar con el método de mejora la solución generada
4. Incluir la solución mejorada en el conjunto P
5. Actualizar las frecuencias almacenadas en c con la nueva solución incluida en P

EJEMPLO: VIAJANTE DE COMERCIO

Medida de distancia

- La medida de distancia se emplea para inicializar, actualizar y reconstruir el conjunto de referencia R
- Dadas dos soluciones s_1 y s_2 , la distancia $d(s_1, s_2)$ entre ellas se calcula de la siguiente manera:

$$d(s_1, s_2) = \frac{\sum_{i=1}^n (f(s_1, s_2, i) + g(s_1, s_2, i))}{2}$$

$$f(s_1, s_2, i) = \begin{cases} 0 & \text{si } s_1[i+1] = s_2[j+1] \\ 1 & \text{si } s_1[i+1] \neq s_2[j+1] \end{cases} \quad g(s_1, s_2, i) = \begin{cases} 0 & \text{si } s_1[i-1] = s_2[j-1] \\ 1 & \text{si } s_1[i-1] \neq s_2[j-1] \end{cases}$$

donde j tal que $s_1[i] = s_2[j]$

- Es decir, se suma el número de aristas diferentes

EJEMPLO: VIAJANTE DE COMERCIO

Resto de componentes

- Método de mejora: búsqueda local del mejor (2-opt)
- Combinación de soluciones: se emplea alguno de los operadores de cruce de orden para AGs (OX, PMX o CX)
- Criterio de parada: número máximo de evaluaciones

BÚSQUEDA DISPERSA

Comentarios Finales

- La Búsqueda Dispersa es un método evolutivo capaz de obtener soluciones de calidad a problemas difíciles
- Algunos aspectos están claramente establecidos y otros requieren de más estudio
- Su arquitectura permite el diseño de sistemas independientes del contexto



METAHEURÍSTICAS

2014 - 2015

- Tema 1. Introducción a las Metaheurísticas
- Tema 2. Algoritmos de Búsqueda Local Básicos
- Tema 3. Métodos Basados en Trayectorias Simples
- Tema 4. Métodos Basados en Trayectorias Múltiples
- Tema 5. Metaheurísticas Basadas en Poblaciones
- Tema 6. Metaheurísticas Basadas en Adaptación Social
- Tema 7. Metaheurísticas Híbridas
- Tema 8. Metaheurísticas Paralelas