

ALGORÍTMICA

2012 - 2013

- **Parte I. Introducción a las Metaheurísticas**
 - Tema 1. Metaheurísticas: Introducción y Clasificación
- **Parte II. Métodos Basados en Trayectorias y Entornos**
 - Tema 2. Algoritmos de Búsqueda Local Básicos
 - Tema 3. Algoritmos de Enfriamiento Simulado
 - Tema 4. Algoritmos de Búsqueda Tabú
 - Tema 5. Métodos Basados en Trayectorias Múltiples I: Métodos Multiarranque Básicos y GRASP
 - Tema 6. Métodos Basados en Trayectorias Múltiples II: ILS y VNS
- **Parte III. Métodos Basados en Poblaciones**
 - Tema 7. Algoritmos Genéticos
- **Parte IV. Intensificación y Diversificación**
 - Tema 8. Estudio del Equilibrio entre Intensificación y Diversificación
- **Parte V. Metaheurísticas Híbridas: Poblaciones y Trayectorias**
 - Tema 9. Algoritmos Meméticos
 - **Tema 10. Scatter Search**
- **Parte VI. Paralelización de Metaheurísticas**
 - Tema 11. Metaheurísticas en Sistemas Descentralizados
- **Parte VII. Conclusiones**
 - Tema 12. Algunas Consideraciones sobre la Adaptación de Metaheurísticas a la Resolución de Problemas

ALGORÍTMICA

Tema 10. Modelos Híbridos II: Búsqueda Dispersa -*Scatter Search*-

- **BÚSQUEDA DISPERSA**
 - **EJEMPLO: VIAJANTE DE COMERCIO**
 - **APLICACIONES**
- 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.

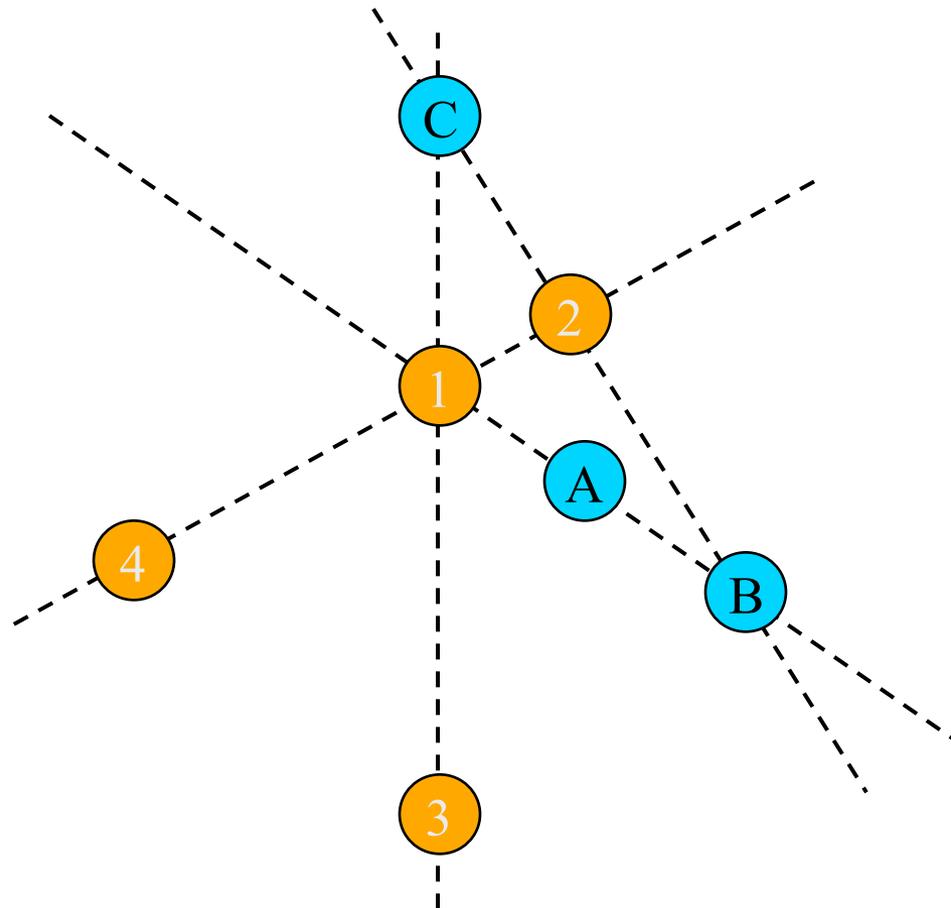
1. BÚSQUEDA DISPERSA

- **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**

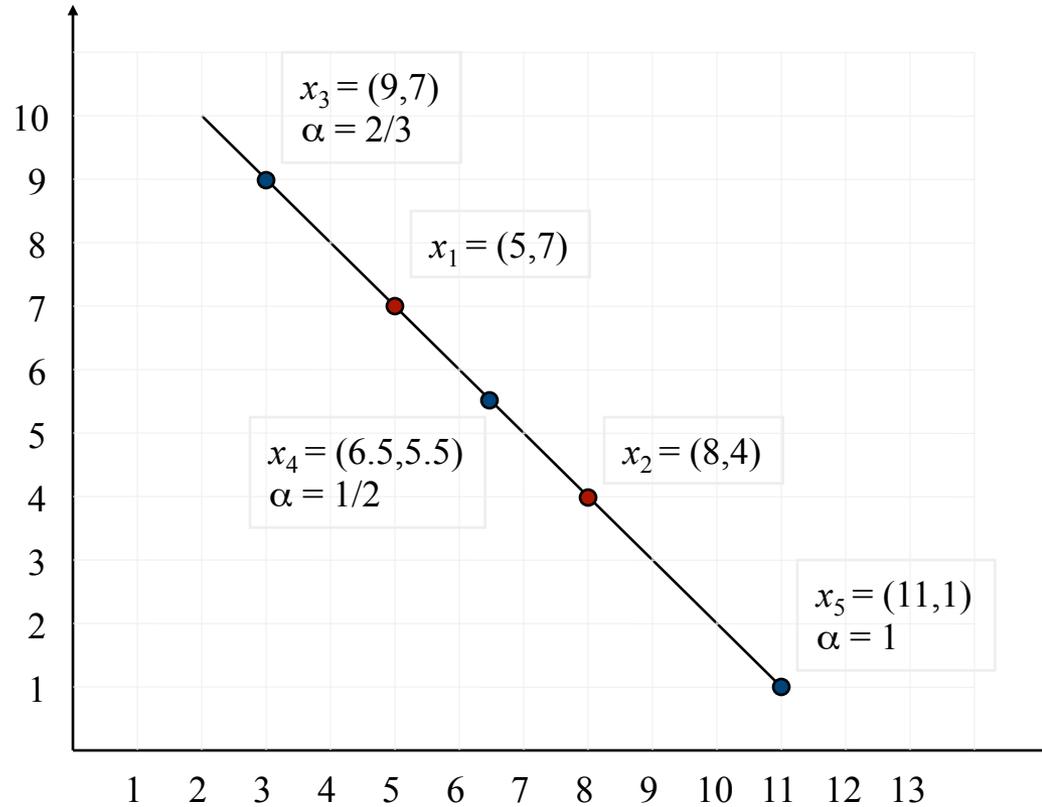
Orígenes

- 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)

Representación Gráfica



Combinación Lineal (2D)



$$x_3 = x_1 - \alpha \cdot (x_2 - x_1)$$

$$x_4 = x_1 + \alpha \cdot (x_2 - x_1)$$

$$x_5 = x_2 + \alpha \cdot (x_2 - x_1)$$

Esquema: “Template”

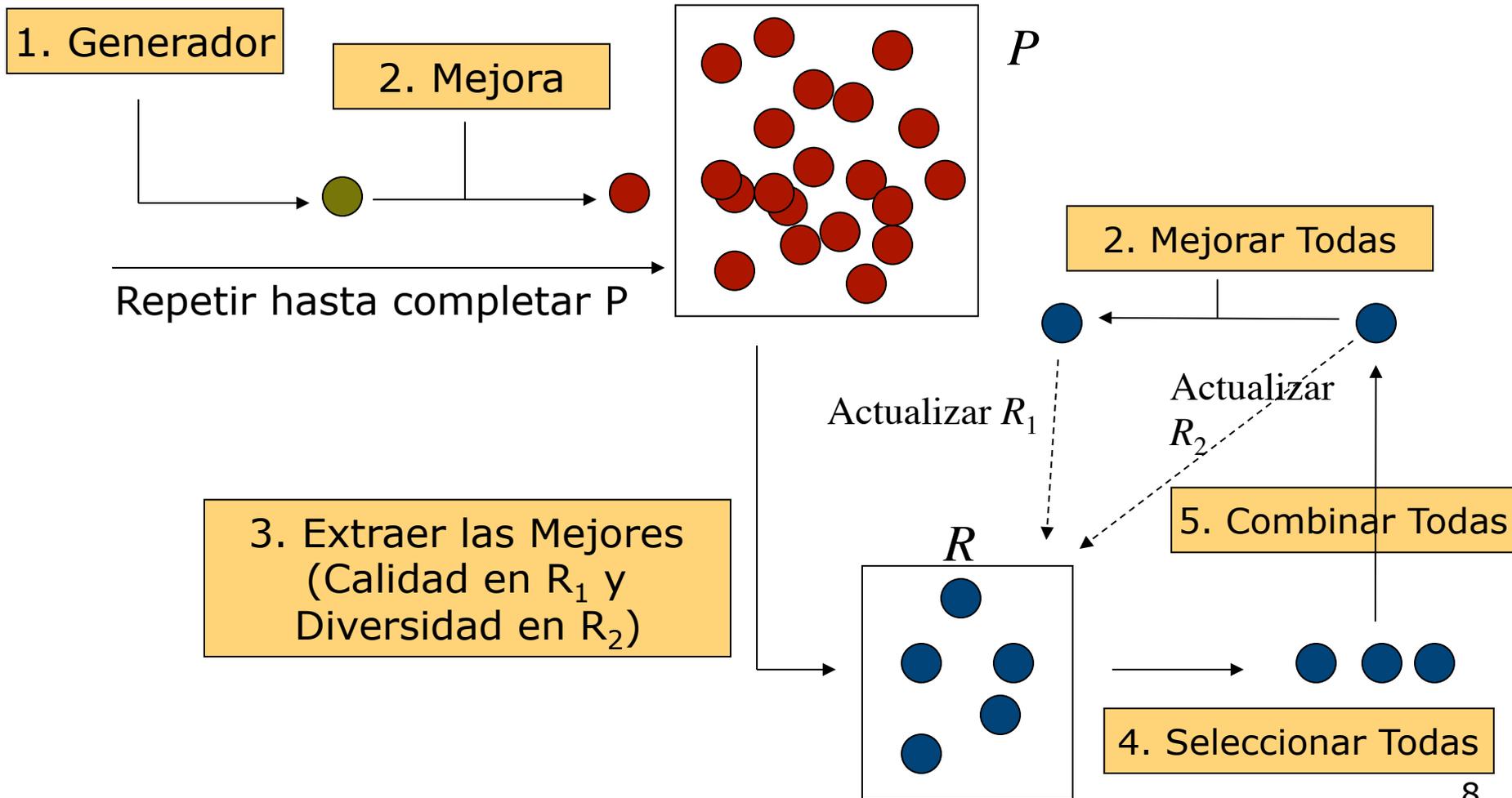
1. Método de Generación Diversa †
2. Método de Mejora ^
3. Conjunto de Referencia †
 - Inicialización
 - Actualización
 - Reconstrucción
4. Generación de Subconjuntos *
5. Combinación de Soluciones ^

† Independiente del Problema (excepto la medida de distancia)

* Independiente del Problema

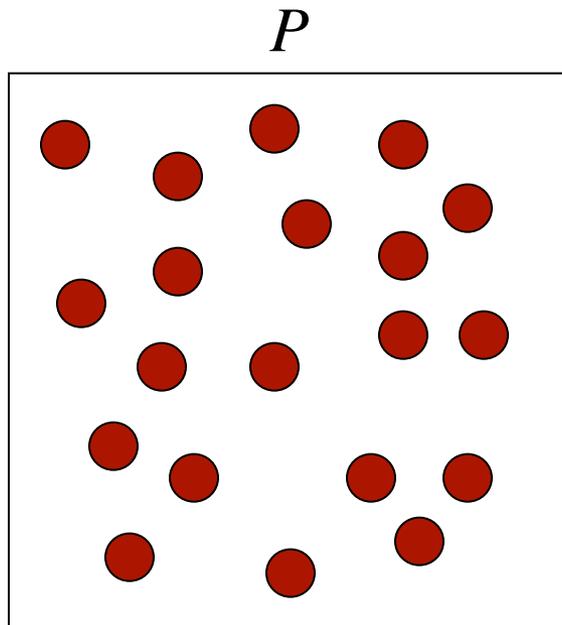
^ Dependiente del Problema

Esquema Básico

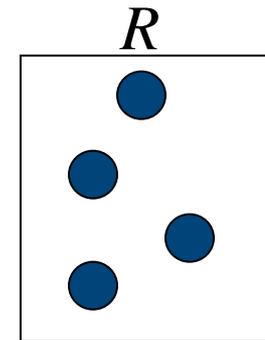


1. Generación Diversa

- 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

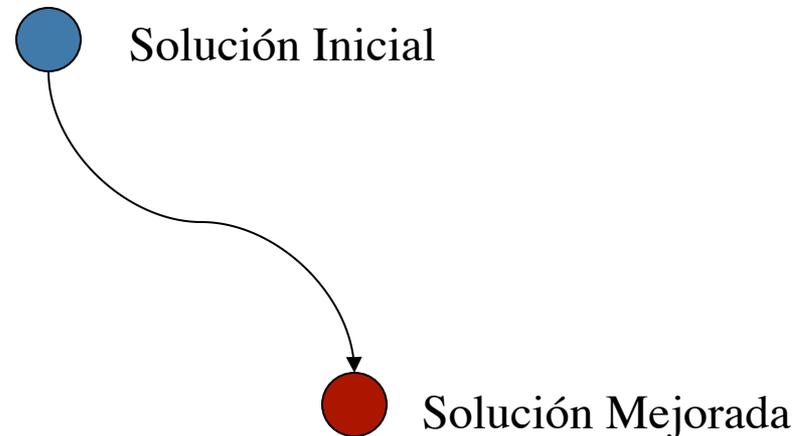


$$|P| \geq 10 \cdot |R|$$



2. Método de Mejora

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

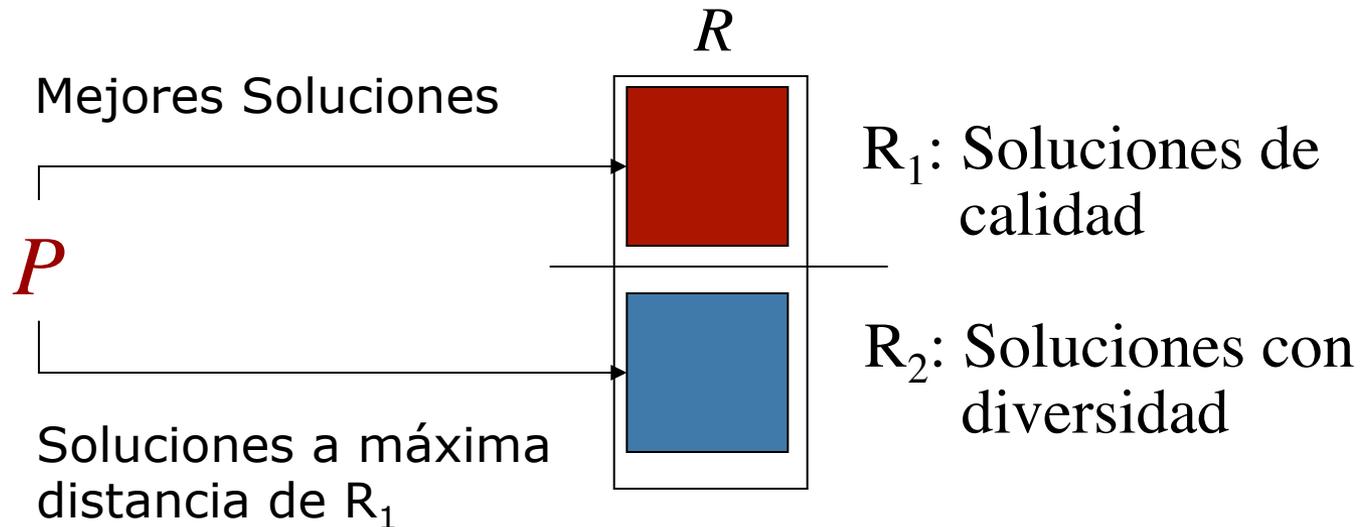


- Se aplica a:
 - Soluciones Originales
 - Soluciones Combinadas

3. Conjunto de Referencia

3.1 Inicialización:

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



3. Conjunto de Referencia

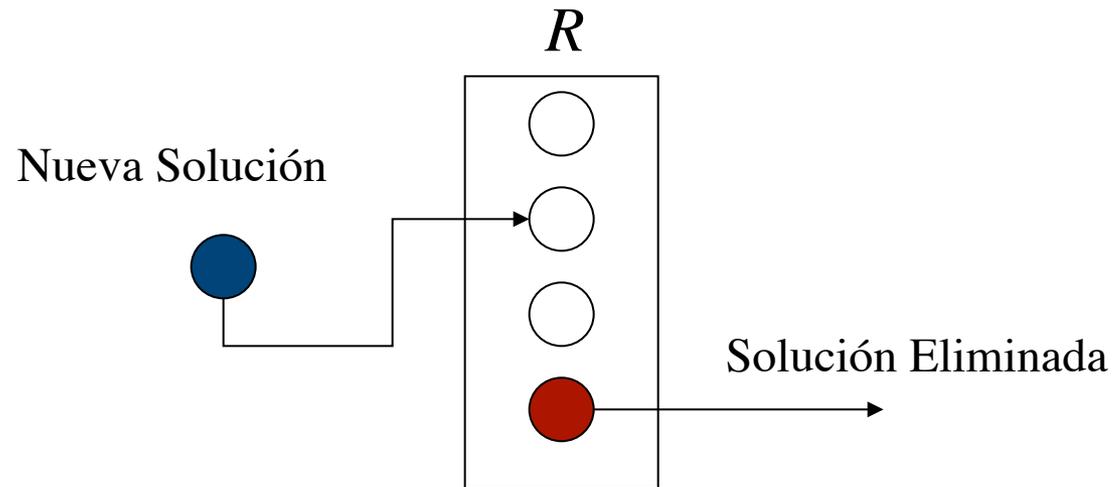
3.1 Inicialización:

- 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

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



3. Conjunto de Referencia

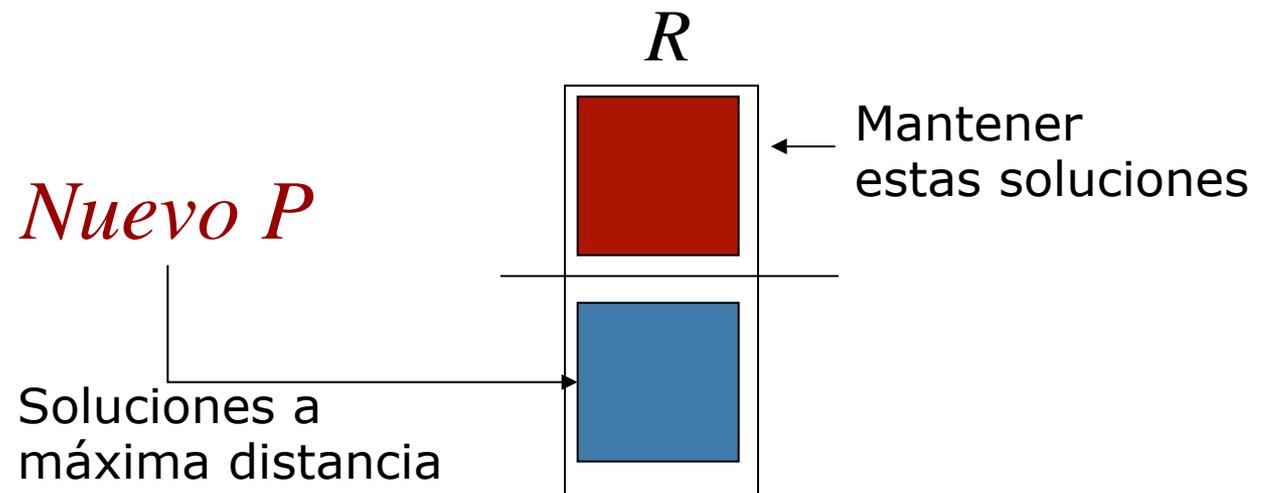
3.2 Actualización:

- 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

3. Conjunto de Referencia

3.3 Reconstrucción

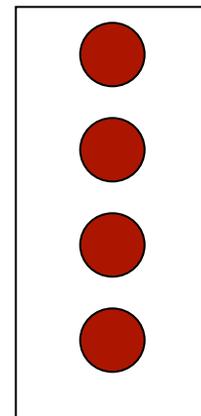
- Cuando en una iteración de la actualización no se consigue mejorar ninguna solución, se reconstruye R
- Mantenemos el subconjunto de calidad (R_1) y eliminamos el de diversidad (R_2)
- Generamos un nuevo P y extraemos $|R_2|$ soluciones diversas respecto a R_1 que pasan a formar el nuevo R_2



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

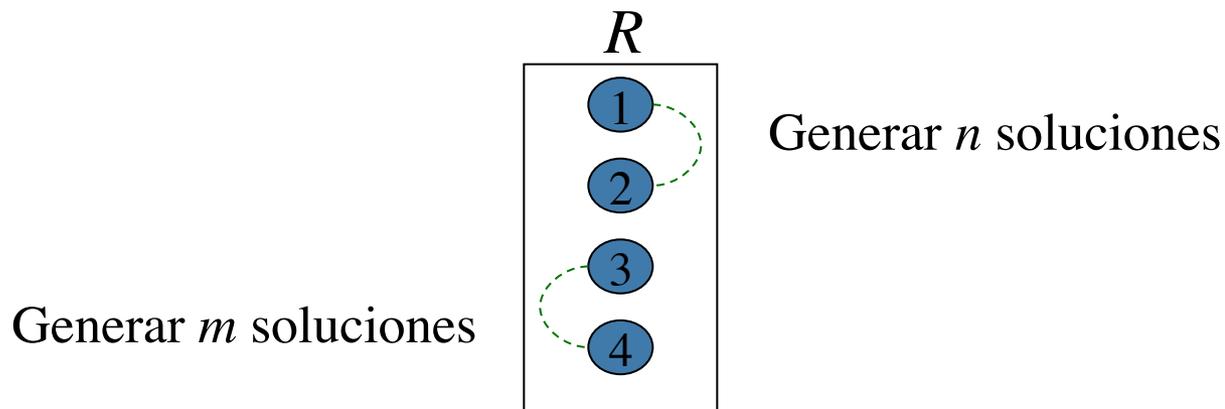


4. Generación de Subconjuntos (Selección)

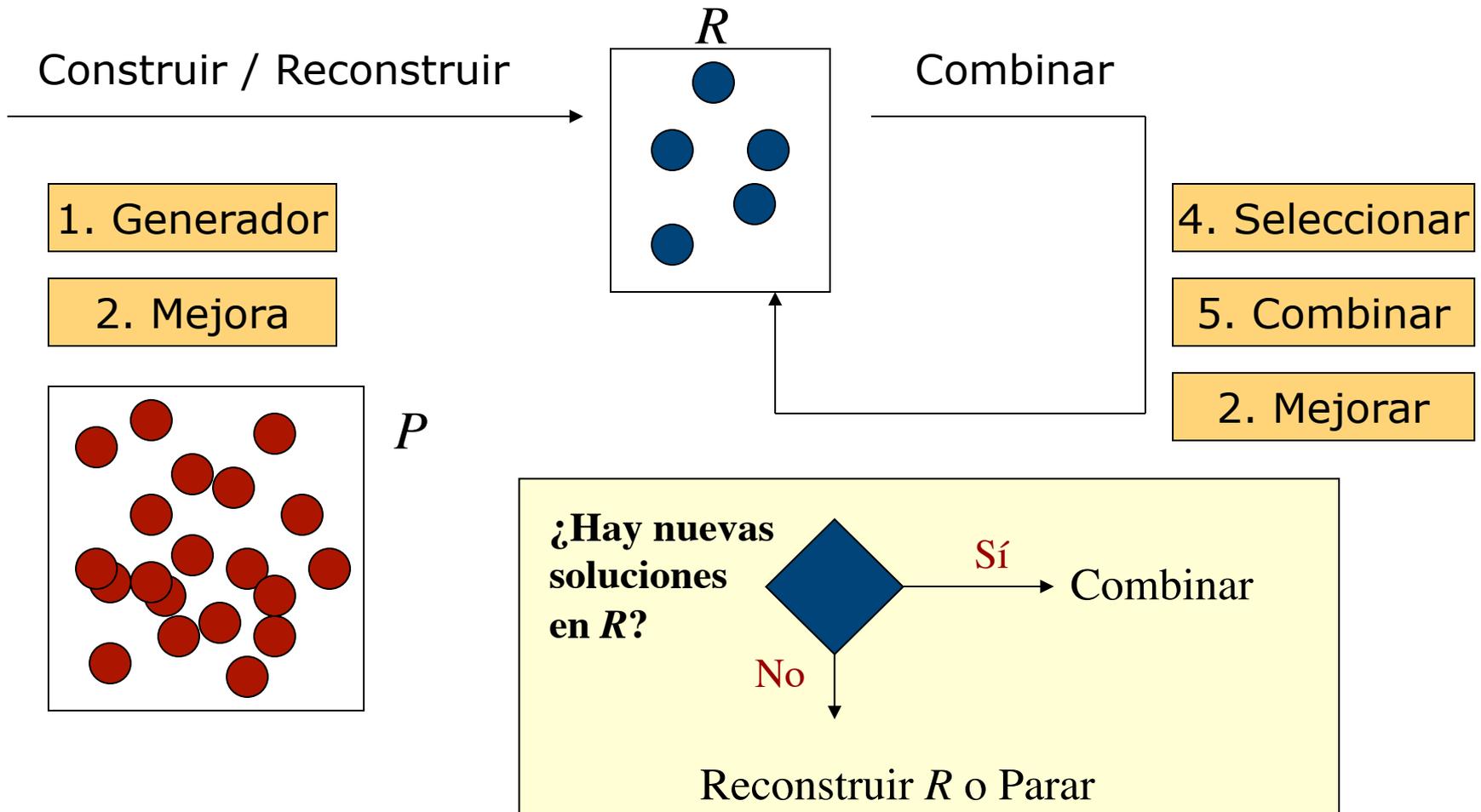
- Por ejemplo, se puede seguir el siguiente procedimiento:
 - Cada subconjunto está compuesto de dos soluciones
 - Se generan todas las parejas posibles teniendo en cuenta que habrá tres grupos:
 - Combinaciones entre dos soluciones de R_1
 - Combinaciones con una solución de R_1 y otra de R_2
 - Combinaciones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

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:



Esquema



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

Código de libre distribución

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

2. 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:

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

2. 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

2. 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

2. 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

3. APLICACIONES

- 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 el número de veces que el patrón i ha sido seleccionado ($S[i]=1$)
 2. Generar una solución “alejada” del conjunto P actual:
 - Hacer un bucle desde el nodo 1 a la n
 - En cada paso, generar el valor 1 con probabilidad:

$$P_i = 1 - c_i / |P|$$

3. APLICACIONES

(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

3. APLICACIONES

- 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 con la métrica de Hamming:

$$d(s_1, s_2) = \sum_{i=1}^n f(s_1, s_2, i), \quad f(s_1, s_2, i) = \begin{cases} 0 & \text{si } s_1[i] = s_2[i] \\ 1 & \text{si } s_1[i] \neq s_2[i] \end{cases}$$

Es decir, se contabilizan los nodos asignados a un subgrupo distinto en cada solución

3. APLICACIONES

- Método de mejora: la búsqueda local de la práctica 1 (con el enfoque del mejor vecino)
- Combinación de soluciones: se emplea el operador de cruce para AGs del Tema 7 (cruce en dos puntos)
- Criterio de parada: número máximo de evaluaciones

CONCLUSIONES

- 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
- Actualmente está en plena expansión

ALGORÍTMICA

2012 - 2013

- **Parte I. Introducción a las Metaheurísticas**
 - Tema 1. Metaheurísticas: Introducción y Clasificación
- **Parte II. Métodos Basados en Trayectorias y Entornos**
 - Tema 2. Algoritmos de Búsqueda Local Básicos
 - Tema 3. Algoritmos de Enfriamiento Simulado
 - Tema 4. Algoritmos de Búsqueda Tabú
 - Tema 5. Métodos Basados en Trayectorias Múltiples I: Métodos Multiarranque Básicos y GRASP
 - Tema 6. Métodos Basados en Trayectorias Múltiples II: ILS y VNS
- **Parte III. Métodos Basados en Poblaciones**
 - Tema 7. Algoritmos Genéticos
- **Parte IV. Intensificación y Diversificación**
 - Tema 8. Estudio del Equilibrio entre Intensificación y Diversificación
- **Parte V. Metaheurísticas Híbridas: Poblaciones y Trayectorias**
 - Tema 9. Algoritmos Meméticos
 - Tema 10. Scatter Search
- **Parte VI. Paralelización de Metaheurísticas**
 - **Tema 11. Metaheurísticas en Sistemas Descentralizados**
- **Parte VII. Conclusiones**
 - Tema 12. Algunas Consideraciones sobre la Adaptación de Metaheurísticas a la Resolución de Problemas