

## Capítulo 4

# Algoritmos Metaheurísticos

### 4.1. Introducción

El término *metaheurística* o *meta-heurística* fue acuñado por F. Glover en el año 1986 [121]. Con este término, pretendía definir un “*procedimiento maestro de alto nivel que guía y modifica otras heurísticas para explorar soluciones más allá de la simple optimalidad local*”.

Actualmente, existe una cantidad muy importante de trabajos científicos publicados que abordan problemas de optimización a través de las metaheurísticas, investigaciones sobre nuevas metaheurísticas o extensiones de las metaheurísticas ya conocidas. Existen bastantes foros donde se publican todos estos trabajos de investigación, entre los que se pueden destacar varias revistas (*Journal of Heuristics* [327] o el *INFORMS Journal of Computing* [328]), libros ([211][318][271][71][133]), congresos temáticos (de ámbito internacional como el *MIC - Metaheuristics International Conference*, o nacional como el *MAEB - Metaheurísticas Algoritmos Evolutivos y Bioinspirados*), sitios web ([329][162]), informes técnicos, números especiales de revistas, etc.

A partir de la definición original de F. Glover, en la literatura se pueden encontrar otras definiciones alternativas de metaheurísticas o heurísticas modernas (como se las denomina en [259][211]), entre las que se pueden destacar las siguientes:

- J.P. Kelly et al. [174]: “*Las metaheurísticas son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Las metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los procedimientos estadísticos*”.

- S. Voss et al. [318]: “Una metaheurística es un proceso iterativo maestro que guía y modifica las operaciones de una heurística subordinada para producir eficientemente soluciones de alta calidad. Las metaheurísticas pueden manipular una única solución completa (o incompleta) o una colección de soluciones en cada iteración. La heurística subordinada puede ser un procedimiento de alto (o bajo) nivel, una búsqueda local, o un método constructivo”.

En [300][329][32] se pueden encontrar otras definiciones del término metaheurística. La idea básica general es siempre la misma: enriquecer a los algoritmos heurísticos de forma que éstos no se queden atrapados en óptimos locales.

La evolución de las metaheurísticas durante los últimos 25 años ha tenido un comportamiento prácticamente exponencial. En el tiempo que transcurre desde las primeras reticencias (por su supuesta falta de rigor científico [96]) hasta la actualidad, se han encontrado soluciones de muy alta calidad a problemas que hace tiempo parecían inabordables.

De modo general, se puede decir que las metaheurísticas combinan ideas que provienen de cuatro campos de investigación bien distintos: las *técnicas de diseño de algoritmos* (resuelven una colección de problemas), *algoritmos específicos* (dependientes del problema que se quiere resolver), *fuentes de inspiración* (del mundo real) y *métodos estadísticos*. En la figura 4.1 se muestra esta idea.

Una primera conclusión que se puede extraer de las definiciones dadas es que, en muchos casos, son reglas de sentido común que permiten hacer una búsqueda “*inteligente*”. Debido a esta característica, para bastantes metaheurísticas no existe un marco teórico que las sustente, sino que es a través de los buenos resultados experimentales donde encuentran su justificación. Aún así, algunos autores han intentado desarrollar un marco matemático donde se pueda establecer, por ejemplo, bajo qué condiciones un procedimiento metaheurístico converge hacia el óptimo. En este sentido, para los algoritmos estocásticos de búsqueda<sup>1</sup>, se ha desarrollado un marco común, conocido como *algoritmos de escalada generalizados* [166][165][171][172], que asegura la convergencia de todos ellos. Dentro de este tipo de algoritmos se incluyen como casos particulares los procedimientos de búsqueda local [2], el recocido simulado [1], los métodos ruidosos [56] y los métodos de aceptación del umbral [91][226]. Se han obtenido resultados semejantes para algoritmos genéticos [287] y búsqueda tabú aleatorizada [98]. Estos desarrollos teóricos no reflejan el comportamiento real de la metaheurística, ya que, aunque aseguran la convergencia en determinadas circunstancias, requieren tiempos exponenciales, que lógicamente, no se corresponden con los requisitos reales.

<sup>1</sup>Procedimientos de búsqueda local con cierta aleatorización en la aceptación de movimientos que empeoran la solución.

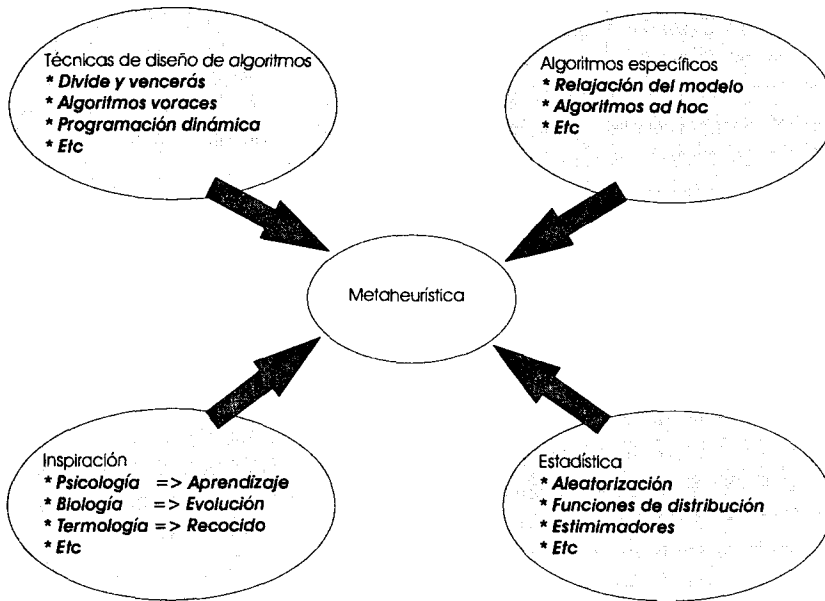


Figura 4.1: Elementos básicos que permiten el diseño de metaheurísticas.

Otro resultado teórico fundamental, en el contexto de los algoritmos genéticos, es el *teorema del esquema* (*schemata theorem*) [160][136]. Los esquemas son una extensión de la codificación binaria de un individuo. Realmente, un esquema representa a un conjunto de individuos con ciertas características comunes (*template*). En este escenario, el teorema del esquema cambia la búsqueda de soluciones por la búsqueda de esquemas, los cuales son instanciados por un individuo, de tal forma que se produce un paralelismo intrínseco en la metaheurística. Desafortunadamente, el teorema NFL [325], que se describirá a continuación, disuadió a los investigadores de seguir por ese camino. El paralelismo intrínseco sólo aparece en algunos problemas.

Es importante resaltar que las metaheurísticas no son la panacea para todos los problemas de optimización combinatoria. En 1997 D. Wolpert y W. Macready [325] probaron que para todas las posibles funciones objetivo<sup>2</sup> todos los algoritmos de búsqueda, tratados como cajas negras, se comportan igual de bien (o de mal), de tal forma que si una técnica es muy eficiente para una colección de problemas, se mostrará muy ineficiente para otra colección. Este resultado, conocido como *teorema NFL* (*No Free Lunch Theorem*) establece que, en media, ni los algoritmos genéticos, ni el re-

<sup>2</sup>Cada función objetivo identifica un problema combinatorio.

cocido simulado, ni la búsqueda tabú ni ninguna metaheurística, es más eficiente que la búsqueda puramente aleatoria.

La conclusión más importante que se puede extraer de este trabajo, no es que no tenga sentido estudiar o diseñar metaheurísticas, sino que en este diseño se debe incluir algún tipo de conocimiento sobre la función objetivo, de tal forma que se pueda responder a la siguiente pregunta: *¿cómo se pueden encontrar soluciones de alta calidad para una función objetivo dada?*. La respuesta a esta pregunta está en el propio artículo de D. Wolpert y W. Macready: determinar ciertas características de la función objetivo (*salient features*) y construir un algoritmo de búsqueda que específicamente encaje con estas características.

La demostración del teorema NFL se basa en la utilización de algoritmos de búsqueda que no visiten dos veces la misma solución. Esta hipótesis, habitualmente, no se verifica en la mayoría de las metaheurísticas. Por ejemplo, en las metaheurísticas poblacionales, de generación a generación se mantienen ciertas estructuras; por lo tanto, se visitarán soluciones ya visitadas. En este sentido, el teorema NFL admite que un algoritmo de búsqueda que utilice estructuras del pasado, que sean adecuadas y que se introduzcan en la siguiente iteración, se comportará mejor que aquél que no lo haga.

Las metaheurísticas deben afrontar este resultado teórico siendo fieles a la descripción que se ha dado al inicio de la sección. En primer lugar, tener una parte general y sencilla, que les permita ser aplicadas a una colección amplia de problemas. En segundo lugar, pueden tener una parte dependiente del problema que les permita obtener un algoritmo eficiente.

Además, las metaheurísticas utilizan como mecanismo para guiar la búsqueda el *principio de optimalidad aproximada* (POP) [127], que se puede enunciar en los siguientes términos:

*“Las soluciones de alta calidad en un nivel determinado, se suelen encontrar muy próximas a las soluciones que también son de alta calidad en un nivel adyacente. Por lo tanto, esta clase de soluciones usualmente tienen estructuras en común”.*

De este principio se puede extraer que, para cualquier problema de optimización, su espacio de búsqueda contiene vecindades en las cuales, las soluciones de alta calidad, están rodeadas de otras soluciones de:

- alta calidad.
- alta y baja calidad.
- mucha peor calidad.

Los procedimientos metaheurísticos se basan también en estas ideas como mecanismo de guía de la búsqueda.

## 4.2. Clasificación de algoritmos metaheurísticos

En los sucesivos capítulos se presentarán algunas de las metaheurísticas que con más éxito se han aplicado a una colección amplia de problemas. Es importante destacar que aunque estas metaheurísticas, a priori, parece que son estrategias de búsqueda muy diferentes, sus principios algorítmicos son bastante parecidos [116]. Durante los últimos años algunos investigadores están desarrollando un *marco general* en el que tengan cabida la mayoría de las metaheurísticas. El desarrollo del marco general viene determinado por el análisis entre las semejanzas y diferencias que existen entre las distintas metaheurísticas. Este análisis es equivalente a encontrar una “*estructura*” que permita organizar las metaheurísticas según una o varias características. Estas características podrían ser: la forma de explorar el espacio de soluciones, el dominio de inspiración, los métodos a partir de los cuales se construye la metaheurística, etc.

La determinación de la estructura del marco general establece una clasificación o taxonomía de las metaheurísticas que, por lo general, es una tarea difícil y arriesgada [185], debido fundamentalmente a que en las implementaciones reales no se pueden extraer un conjunto de características que sean totalmente ortogonales y que permitan una separación completa entre unas metaheurísticas y otras. Una metaheurística tiene dos interpretaciones distintas [26][32]. La primera de ellas hace referencia al “*estado puro*” [26] de la metaheurística. La segunda visión considera la implementación real, que habitualmente incluye “*inhomogeneidades*” [26][32] que la suelen hacer más eficiente. Debido a estas inhomogeneidades la clasificación de las metaheurísticas se complica enormemente. Con el objetivo de obtener una clasificación más clara y útil, se han considerado exclusivamente las características conceptuales de las metaheurísticas (estado puro).

A partir de las taxonomías que aparecen en la literatura, se proponen varias alternativas que permiten clasificar las metaheurísticas revisadas en este capítulo.

### 4.2.1. Taxonomías clásicas

En la literatura se pueden encontrar diferentes formas de describir o clasificar las metaheurísticas, dependiendo fundamentalmente de las características que se seleccionen para dicho propósito. A continuación, se describen brevemente algunas de las formas que más comúnmente se han utilizado para clasificar metaheurísticas [32]:

- **Atendiendo a la Inspiración:**

*Natural:* algoritmos que se basan en un símil real, ya sea biológico, social, cultural, etc.

*Sin inspiración:* algoritmos que se obtienen directamente de sus propiedades matemáticas.

- **Atendiendo al número de soluciones:**

*Poblacionales:* buscan el óptimo de un problema a través de un conjunto de soluciones.

*Trayectoriales:* trabajan exclusivamente con una solución que mejoran iterativamente.

- **Atendiendo a la función objetivo:**

*Estáticas:* no hacen ninguna modificación sobre la función objetivo del problema.

*Dinámicas:* modifican la función objetivo durante la búsqueda.

- **Atendiendo a la vecindad:**

*Una vecindad:* durante la búsqueda utilizan exclusivamente una estructura de vecindad.

*Varias vecindades:* durante la búsqueda modifican la estructura de la vecindad.

- **Atendiendo al uso de memoria:**

*Sin memoria:* se basan exclusivamente en el estado anterior.

*Con memoria:* utilizan una estructura de memoria para recordar la historia pasada.

Cualquiera de las alternativas descritas por sí solas no es de grano suficientemente fino como para permitir una separación clara entre todas las metaheurísticas. Generalmente, estas características (pueden incluirse más) se suelen combinar para permitir una clasificación más elaborada.

#### 4.2.2. Taxonomía tabular

La estructuración tabular, como mecanismo de análisis de las diferencias y semejanzas entre las distintas metaheurísticas es, probablemente, la taxonomía más usada y más sencilla. Consiste en construir una tabla de características que permita la comparación entre metaheurísticas. Esta tabla tiene una columna por cada una de las características descritas anteriormente (aunque podrían incluirse más). Cada celda de

Paradigma	Implementación	Inspiración	Multi- arranque	Búsqueda local	Solución inicial	Función objetivo	Niveles de vecindad	Vecindad	Memoria	Proceso aleatorio	Procesos adaptativos
Optimización por colonias de hormigas - ACO	eco tradicional eco + demonios	si	no	no si	poblac.	estática dinámica	uno	estática	explícita	no	si
Equipos síncronos - AT	AT	parcial	no	si	poblac.	estática	varios	estática	explícita	no	si
Algoritmos Culturales - CA	CA	si	no	no	poblac.	estática	varios	dinámica	explícita	si	si
Algoritmos de estimación de la distribución - EDA	EDA	parcial	no	no	poblac.	estática	varios	*	implícita	no	si
Búsqueda por entorno adaptativo borroso - FANS	FANS	no	no	si	trayec.	dinámica	varios	estática	no	no	si
Algoritmos Genéticos - GA	GA	si	no	no	poblac.	estática	varios	estática	implícita	si	si
Proc. aleatorizados y adaptativos de búsqueda voraz - GRASP	GRASP tradicion GRASP biased	no	si	si	trayect.	estática	varios	dinámica	no	si	no
Búsqueda local guiada - GLS	GLS	no	no	si	trayect.	dinámica	uno	estática	explícita	no	no
Concentración heurística - HC	HC	no	si	si	trayect.	estática	varios	estática	explícita	no	no
Búsqueda local iterativa - ILS	ILS tradicional ILS reactivo	no	no	si	trayect.	estática	varios	estática	no	si	no si
Algoritmos Meméticos - MA	MA	parcial	no	si	poblac.	estática	varios	estática	implícita	si	si
Métodos Multi-arranque - MSM	MSM básico AMS	no	si	si	trayect.	estática	uno	estática	no	si no	no si
Métodos ruidosos - NM	NN-ruido datos NN-ruido f NN-perturbación	no	no	si	trayect.	dinámica dinámica estática	uno	estática	no	si	no no no
Metahe. de opt. parcial en condiciones. esp. de in- tensificación - POPMUSIC	POPMUSIC	no	no	si	trayect.	estática	uno	estática	no	no	no
Reencadenamiento de trayectorias - PR	PR	no	*	si	poblac.	estática	uno	estática	implícita	no	si
Recocido simulado - SA	SA	si	no	si	trayect.	estática	uno	estática	no	si	no
Inteligencia de enjambre - SI	SI	si	no	no	poblac.	dinámica	uno	dinámica	implícita	si	si
Búsqueda dispersa - SS	SS	no	no	si	poblac.	estática	varios	estática	explícita	no	si
Métodos de aceptación del umbral - TAM	TAM	no	no	si	trayect.	estática	uno	estática	no	no	no
Búsqueda tabú - TS	TS tradicional TS aleatorio TS reactivo	no	no	si	trayect.	estática	uno	dinámica	explícita	no si no si	no no no si
Búsqueda en vecindad variable - VNS	VND RVNS BVNS	no	no	si no si	trayect.	estática	varios	estática	no	no si si	no no no

Figura 4.2: Resumen de las características que presentan las distintas metaheurísticas.

la tabla se rellenaría con el valor correspondiente. Por ejemplo: en el caso de la búsqueda tabú, la celda correspondiente a la memoria debería estar marcada; en cambio, la celda que se corresponde con poblaciones de soluciones no estaría marcada.

En [185][26][48][32][116] se pueden encontrar ejemplos de este tipo de clasificación. Basándose en estos trabajos, a continuación se muestran las características que se han considerado más descriptivas para realizar una clasificación tabular:

- **Inspiración:** determina si una metaheurística se inspira en un símil real o no. El valor de esta casilla será: “sí” en caso afirmativo, “no” en caso negativo y “parcialmente” en el caso de que se base en una metaheurística con inspiración

real, aunque no se inspire directamente. Por ejemplo, los algoritmos genéticos están basados en las teorías neo-darwinistas de evolución de las especies.

- **Número de soluciones:** determina la cantidad de soluciones que utiliza la metaheurística para implementar la búsqueda. El valor de esta casilla puede ser “trayectorial” en el caso de utilizar una única solución y “poblacional” en el caso de que se utilice un conjunto de soluciones.
- **Multi-arranque:** determina si la metaheurística se ejecuta una o varias veces. El valor de esta casilla será “sí” en el caso de que sea multi-arranque, y “no” en caso contrario.
- **Búsqueda local:** determina si la metaheurística utiliza un procedimiento de búsqueda local para mejorar las soluciones. El valor de esta casilla será “sí” en caso afirmativo, y “no” en caso negativo.
- **Función objetivo:** determina si la metaheurística modifica la función objetivo para hacer la búsqueda o no. El valor de esta casilla será “dinámica” en el caso de que se modifique la función objetivo y “estática” en el caso contrario.
- **Niveles de vecindad:** determina si la metaheurística utiliza una o varias estructuras de vecindad. En el primer caso el valor de la casilla será “uno” y en el segundo caso “varios”.
- **Vecindad:** establece si la metaheurística utiliza una estructura de vecindad estática o dinámica. En el primer caso el valor de esta casilla será “estática” y en el segundo caso “dinámica”. En el caso de los EDAs, no tiene sentido hablar de vecindad
- **Memoria:** establece si la metaheurística utiliza estructura de memorias (en cualquiera de sus implementaciones) para hacer la búsqueda. El valor de esta casilla será “explícita” en el caso de tener una estructura explícita que almacene y guíe la búsqueda o “implícita” cuando se tienen en cuenta características del pasado, pero no de forma explícita. No se considera memoria al almacenamiento de información, si éste no se utiliza para guiar la búsqueda. Por ejemplo, GRASP almacena la mejor solución encontrada en cada iteración, pero a no ser que sea reactivo, esta memoria no guía la búsqueda.
- **Procesos aleatorios:** determina si la metaheurística contempla fenómenos de aleatorización para permitir movimientos que puedan empeorar la solución para salir de óptimos locales. El valor de la casilla será “sí” en caso afirmativo y “no” en el caso contrario.
- **Procesos adaptativos:** determina si la metaheurística se adapta al problema, de tal forma que es capaz de cambiar sus parámetros, ya sea para intensificar o



para diversificar la búsqueda. El valor de la casilla será “sí” en el caso de que algún procedimiento sea adaptativo o “no” en caso negativo. Esta característica se debe considerar conjuntamente con otras características como la búsqueda local, multi-arranque o el proceso constructivo.

Para tener una clasificación más clara, se ha hecho una separación entre lo que es el paradigma de la metaheurística y la implementación concreta. Por ejemplo, se considera que GRASP es el paradigma de la metaheurística que lleva su propio nombre y “GRASP tradicional” o “GRASP reactivo” son dos implementaciones concretas de esa metaheurística. En la figura 4.2 se muestra una clasificación tabular para las metaheurísticas descritas en este capítulo.

### 4.2.3. Taxonomía jerárquica

La clasificación jerárquica es bastante más compleja que la clasificación tabular, debido fundamentalmente a que la primera clasificación debe describir de forma explícita la estructura que relaciona todas las metaheurísticas, mientras que en la segunda clasificación, esta estructura es implícita. Además, la clasificación jerárquica supone que no existe solapamiento entre las metaheurísticas, hipótesis que por lo general no tiene por qué ser cierta. Por ejemplo, no es cierto para metaheurísticas que hereden características de más de una (hibridación de metaheurísticas).

Las taxonomías jerárquicas más sencillas establecen una estructura de árbol para intentar clasificar las metaheurísticas. Estas clasificaciones asumen que son parciales, ya que la relación entre estos algoritmos suele ser más compleja. Aún así, esta clasificación es bastante útil, debido a que con muy pocas clases se pueden clasificar muchas metaheurísticas. Ejemplos de este tipo de jerarquías se pueden encontrar en [154][79].

En la figura 4.3 se presenta una propuesta de taxonomía que permite clasificar jerárquicamente la mayoría de las metaheurísticas revisadas en este capítulo. Naturalmente, esta taxonomía presenta problemas, ya que probablemente algunas de las clases se solapen entre sí. La característica más importante de la taxonomía jerárquica es que permite agrupar las metaheurísticas desde un punto de vista conceptual.

Es importante destacar que la jerarquía propuesta no puede clasificar todas las metaheurísticas revisadas en este libro. Por ejemplo, la búsqueda tabú aleatorizada [98] estaría a medio camino entre las metaheurísticas basadas en búsqueda local y las metaheurísticas basadas en búsquedas aleatorizadas. Para tener una descripción más detallada se debe recurrir a una estructura de grafo, más compleja que la de árbol. En este campo, el desarrollo de librerías de clases es, probablemente, desde donde más esfuerzos se ha hecho para conseguir clasificar de forma completamente ortogonal cada una de las metaheurísticas. Así, se diseña una *jerarquía de clases* que permita

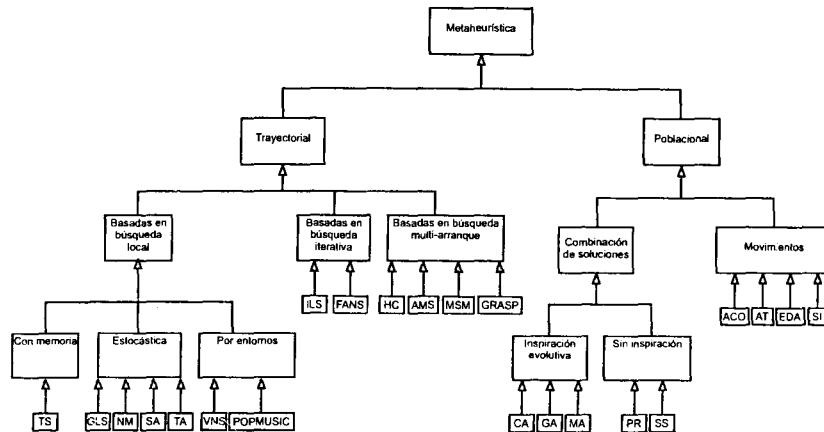


Figura 4.3: Estructura de árbol que permite la clasificación jerárquica de las metaheurísticas.

construir una metaheurística concreta. La jerarquía debe ser lo suficientemente flexible como para construir una metaheurística y alguna de sus variantes. Además, si la jerarquía de clases está bien diseñada, no se produce ninguna intersección entre cada una de las clases. En [326][105] se pueden encontrar taxonomías de este tipo.

La jerarquía propuesta en la figura 4.3 permite una clasificación conceptual de las metaheurísticas, pero no da ninguna información acerca de las características algorítmicas de cada una de ellas. En [116] se propone un marco que permite describir algorítmicamente la mayoría de las metaheurísticas. Para ello, supone que todas las metaheurísticas están compuestas por un conjunto relativamente pequeño de métodos. A continuación, se describen los métodos utilizados en este libro:

- **Construcción:** lo emplean todas las metaheurísticas para crear una o un conjunto de soluciones. Además, como se verá en capítulos sucesivos, en el caso de GRASP y ACO utilizan este método para construir soluciones iniciales de forma iterativa.
- **Recombinación:** permite generar nuevas soluciones a partir de las actuales por medio del cruce, en el sentido amplio del término. Este método sólo lo utilizan algunas metaheurísticas poblacionales evolutivas.
- **Modificación aleatoria:** se utiliza para introducir una perturbación aleatoria que modifica una o varias soluciones actuales.

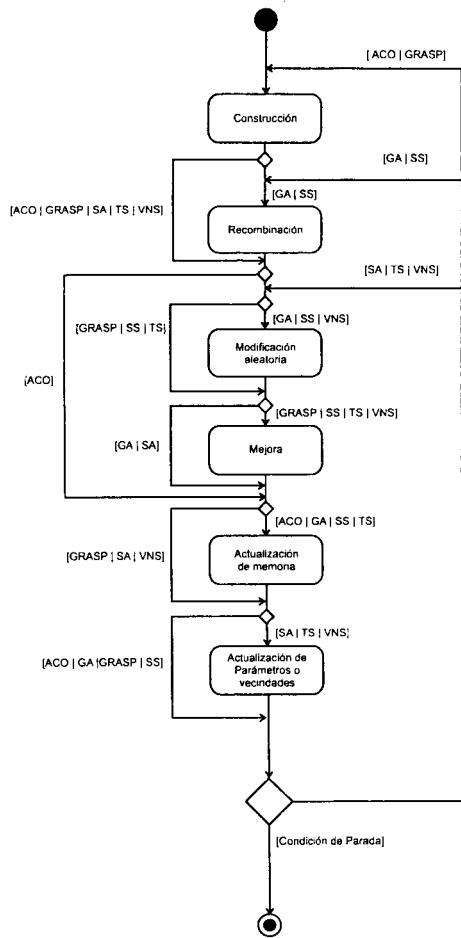


Figura 4.4: Vista algorítmica unificada de las metaheurísticas.

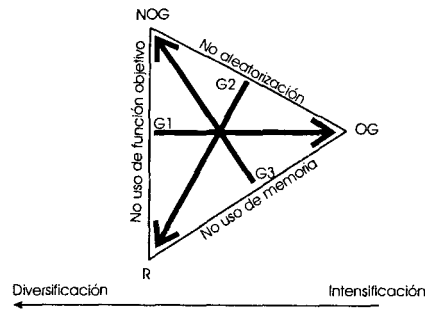


Figura 4.5: Espacio intensificación-diversificación.

- **Mejora:** se emplea para mejorar una o varias soluciones. Se puede implementar mediante un procedimiento de búsqueda local o incluso a través de una metaheurística.
- **Actualización de memoria:** se encarga de actualizar la memoria, tanto explícitamente (p.ej. listas tabú en la búsqueda tabú o rastro de feromonas en la colonia de hormigas) como implícitamente (p.ej. poblaciones en algoritmos genéticos o conjunto de referencia en la búsqueda dispersa).
- **Actualización de parámetros o vecindades:** se utiliza para actualizar los parámetros de la búsqueda (modificación de los criterios de aceptación, reducción del ruido) y las estructuras de las vecindades, como en VNS.

En la figura 4.4, se presenta un diagrama algorítmico que ilustra una visión unificada y general de las metaheurísticas basada en la aplicación de las posibles combinaciones de los métodos descritos anteriormente. En esta figura se puede observar que la descripción algorítmica de cada una de las metaheurísticas es un camino entre el nodo *Inicio* y el nodo *Fin*.

#### 4.2.4. Taxonomía basada en la relación intensificación-diversificación

Otra alternativa considerada para clasificar las metaheurísticas consiste en evaluar cómo efectúan el proceso de búsqueda cada una de ellas. Este hecho se manifiesta en cómo hacen el balance entre la intensificación y la diversificación (ver sección 1.2.2).

En este tipo de clasificaciones, se supone que todos los métodos de las metaheurísticas tienen un efecto directo en la intensificación y/o en la diversificación. Como ejemplos de métodos que afectan a la intensificación y a la diversificación se pueden destacar: el criterio de aceptación y la cola de enfriamiento en recocido simulado; la selección, recombinación y mutación en algoritmos evolutivos; la perturbación, la búsqueda local y el criterio de aceptación en la búsqueda local iterativa; etc. Todos estos métodos se analizarán con detalle en capítulos sucesivos. En [32] se propone el análisis de estos métodos como mecanismo para extraer las semejanzas y diferencias entre las distintas metaheurísticas. Para ello, utilizan un espacio bidimensional que permite situar cada uno de estos métodos en un dominio, caracterizado por restringir a los valores de dichos métodos dentro de una superficie triangular, cuyos vértices se corresponden con tres casos extremos de relaciones intensificación-diversificación. En la figura 4.5 se muestra un ejemplo de este triángulo.

El vértice etiquetado con *OG* se corresponde con aquellos métodos que exclusivamente se guían por la función objetivo del problema. Por ejemplo, un procedimiento de búsqueda local aislado y sin ningún tipo de aleatorización estaría muy cerca de este vértice. El vértice *NOG* engloba a aquellos métodos que utilizan una o más funciones de guía pero que no son la función objetivo del problema. Por ejemplo, un procedimiento multi-arranque determinista sin aleatorización estarían muy cerca de este vértice. Por último, el vértice *R* determina a aquellos procedimientos que no tienen ninguna función de guía, o lo que es lo mismo, que son totalmente aleatorios.

La esquina *OG* se corresponde con los métodos de las metaheurísticas que tienen máxima intensificación y mínima diversificación. En cambio, las esquinas *NOG*, *R* y el segmento que la une se corresponden con los métodos de las metaheurísticas que tienen máxima diversificación y mínima intensificación.

En [32] se describe que, sobre el triángulo definido, se pueden establecer tres gradientes diferentes:

- G1: un método cuanto menos utiliza la función objetivo más lejos está del vértice *OG*.
- G2: un método cuanto menos utiliza la aleatorización más lejos está del vértice *R*.
- G3: un método cuanto menos utiliza funciones alternativas a la función objetivo, generalmente relacionadas con el uso de memorias explícitas (búsqueda tabú) o implícitas (algoritmos poblacionales), más lejos estará de *NOG*.

En el triángulo descrito en esta sección se sitúan de forma sencilla aquellas metaheurísticas cuyos métodos tienen bien separada la parte de intensificación de la parte de diversificación. Por ejemplo, desde un punto de vista intuitivo, la búsqueda tabú

es un procedimiento de búsqueda local que tiene restringidos algunos movimientos por una o varias listas tabú (se puede encontrar una descripción detallada del método en la sección 5.2). Esta forma de explorar el espacio de búsqueda tiene efectos tanto en la intensificación como en la diversificación. El hecho de evitar que se accedan a todos los vecinos favorece la intensificación, mientras que el hecho de elegir al mejor vecino disponible favorece la intensificación. Estos dos efectos se pueden controlar a través de la longitud de la lista tabú, de tal forma que las listas largas favorecen la diversificación y las listas cortas favorecen la intensificación. Con respecto al triángulo descrito, se tendrá que el método estará situado en el segmento *NOG – OG*, de tal forma que cuanto más corta sea la lista tabú más cerca estará el algoritmo de *OG*.

### 4.3. Limitaciones de los algoritmos metaheurísticos

Según el teorema NFL, los métodos generales de búsqueda, entre los que se encuentran las metaheurísticas, se comportan exactamente igual cuando se promedian sobre todas las funciones de objetivo posibles, de tal forma que si un algoritmo *A* es más eficiente que un algoritmo *B* en un conjunto de problemas, debe existir otro conjunto de problemas de igual tamaño para los que el algoritmo *B* sea más eficiente que el *A*. Esta aseveración establece que, en media, ninguna metaheurística (algoritmos genéticos, búsqueda dispersa, búsqueda tabú, etc.) es mejor que la búsqueda completamente aleatoria. Una segunda característica que presentan las metaheurísticas es que existen pocas pruebas sobre su convergencia hacia un óptimo global; es decir, que a priori no se puede asegurar ni que la metaheurística converja ni la calidad de la solución obtenida. Por último, las metaheurísticas más optimizadas son demasiado dependientes del problema o al menos necesitan tener un elevado conocimiento heurístico del problema. Esto hace que, en general, se pierda la genericidad original con la que fueron concebidas.

A pesar de estos aparentes problemas, la realidad es que el comportamiento experimental de la mayoría de las metaheurísticas es extraordinario, convirtiéndose para muchos problemas difíciles de resolver en la única alternativa factible para encontrar una solución de calidad en un tiempo razonable. En general, las metaheurísticas se comportan como métodos muy robustos y eficientes que se pueden aplicar con relativa facilidad a una colección amplia de problemas. Además, la demostración del teorema NFL se basa en que el algoritmo de búsqueda no visita dos veces la misma solución y en que no se introduce conocimiento heurístico en el diseño del método metaheurístico. Estas hipótesis habitualmente no son ciertas.