

Metaheurísticas: una revisión actualizada*

Julio Brito Santana, Clara Campos Rodríguez,
Félix C. García López, Miguel García Torres,
Belén Melián Batista, José A. Moreno Pérez,
J. Marcos Moreno Vega.

Grupo de Computación Inteligente

Universidad de La Laguna

gci@ull.es

<http://webpages.ull.es/users/gci/>

Documento de Trabajo nº 02/2004

Departamento de Estadística, Investigación Operativa y Computación.
Universidad de La Laguna. 38271 La Laguna.

20 de junio de 2004

Resumen

Las metaheurísticas pueden concebirse como estrategias generales de diseño de procedimientos heurísticos para la resolución de problemas con un alto rendimiento. En este trabajo se tratan, en primer lugar, los fundamentos para establecer el concepto de metaheurística y apreciar su relevancia actual. La relevancia de las metaheurísticas se aprecia en las publicaciones aparecidas y en la existencia de otras fuentes de documentación disponibles en la actualidad. Las estrategias metaheurísticas se refieren al diseño de alguno de los tipos fundamentales de procedimientos heurísticos de solución de un problema de optimización. Se realiza una descripción de las principales metaheurísticas para los tipos fundamentales de procedimientos alternativos de solución de problemas de optimización: los métodos de relajación, los procesos constructivos, las búsquedas por entornos y los procedimientos evolutivos. Se presta atención especial a las metaheurísticas de búsqueda, que constituyen el paradigma central de estas técnicas, y se incluye una breve descripción de las técnicas tratando de cubrir todo el espectro de propuestas realizadas. Se proponen y analizan las características deseables de las metaheurísticas, desde el punto de vista de su estudio teórico y de su aplicación práctica. Finalizamos con unas breves conclusiones derivadas de nuestra perspectiva.

*Este trabajo ha sido parcialmente financiado con el proyecto TIC2002-04242-C03-01 con fondos FEDER en un 70 %

Índice

1. Introducción	4
2. Las metaheurísticas	4
2.1. Concepto de metaheurística	4
2.2. Tipos de metaheurísticas	6
2.2.1. Metaheurísticas de relajación	7
2.2.2. Metaheurísticas constructivas	8
2.2.3. Metaheurísticas de búsqueda	8
2.2.4. Metaheurísticas evolutivas	9
2.2.5. Otros tipos de metaheurísticas	9
2.3. Relevancia de las metaheurísticas	10
2.3.1. Libros sobre metaheurísticas	10
2.3.2. Artículos sobre metaheurísticas	10
2.3.3. Enlaces relacionados	11
2.3.4. Software sobre metaheurísticas	12
3. Metaheurísticas de búsqueda	14
3.1. Estructura de entornos	16
3.2. Búsquedas locales	18
3.2.1. Búsquedas no informadas	18
3.2.2. Búsquedas locales monótonas	19
3.3. Búsquedas globales	21
3.3.1. Búsqueda multi-arranque	21
3.3.2. Búsqueda por entornos variables	22
3.3.3. Búsquedas no monótonas	22
3.4. Búsquedas basadas en poblaciones o evolutivas	23
3.4.1. Algoritmos genéticos	23
3.4.2. Algoritmos de estimación de distribuciones	24
3.4.3. Búsqueda dispersa	24
3.4.4. Búsqueda por reencadenamiento de camino	25
3.4.5. Algoritmos bionómicos	25
3.5. Otras metaheurísticas de búsqueda	26
3.5.1. Optimización de partículas inteligentes	26
3.5.2. Búsqueda por ruido	26
3.5.3. Búsqueda local iterada	27
3.5.4. Heurística de concentración	27
3.5.5. Búsqueda local guiada	27
3.5.6. Metaheurística FANS	27
3.5.7. Búsqueda reactiva	28
3.5.8. Algoritmos Meméticos	28
4. Otras metaheurísticas	28
4.1. Redes neuronales	29
4.2. Colonias de hormigas	29
4.3. Optimización extrema	30
4.4. Metaheurísticas GRASP	30
4.5. Satisfacción de restricciones	30
4.6. Metaheurísticas multiobjetivo	31

4.7. Metaheurísticas paralelas	31
4.8. Hiperheurísticas	31
5. Propiedades deseables	31
6. Conclusiones	34

1. Introducción

En Inteligencia Artificial (IA) se emplea el calificativo *heurístico*, en un sentido muy genérico, para aplicarlo a todos aquellos aspectos que tienen que ver con el empleo de conocimiento en la realización dinámica de tareas. Se habla de heurística para referirse a una técnica, método o procedimiento inteligente de realizar una tarea que no es producto de un riguroso análisis formal, sino de conocimiento experto sobre la tarea. En especial, se usa el término heurístico para referirse a un procedimiento que trata de aportar soluciones a un problema con un buen rendimiento, en lo referente a la calidad de las soluciones y a los recursos empleados.

En la resolución de problemas específicos han surgido procedimientos heurísticos exitosos, de los que se ha tratado de extraer lo que es esencial en su éxito para aplicarlo a otros problemas o en contextos más extensos. Diversas líneas de investigación de la IA, han contribuido al desarrollo científico del campo de las heurísticas y a extender la aplicación de sus resultados, en especial los sistemas expertos. De esta forma se han obtenido, tanto técnicas y recursos computacionales específicos, como estrategias generales de diseño para procedimientos heurísticos de resolución de problemas. Estas estrategias generales para construir algoritmos, que quedan por encima de las heurísticas, y van algo más allá, se denominan metaheurísticas. Las metaheurísticas pueden integrarse como un sistema experto para facilitar su uso genérico a la vez que mejorar su rendimiento.

En este trabajo se presenta una visión global actualizada del campo de las metaheurísticas, centrada en torno a la noción de metaheurística, la clasificación de las más relevantes y el análisis de las cualidades deseables de éstas. A la que han contribuido diversos autores con reflexiones intercaladas en libros o artículos sobre metaheurísticas específicas (ver, por ejemplo, [GL97], [HM03], [HM02] y [YI01]). Sin embargo, dado su amplitud y lo cambiante de sus fronteras, el estudio completo y riguroso del concepto de metaheurística, una clasificación estructurada y exhaustiva de las diferentes estrategias, y de las características apropiadas de una metaheurística es una empresa casi imposible de contemplar.

En la siguiente sección se describen los fundamentos que permiten establecer, partiendo de la noción de heurística, el concepto de metaheurística y se establece una primera clasificación de las metaheurísticas a partir de los diferentes tipos de procedimientos heurísticos para los que establecen pautas de diseño. En la tercera sección se describen las metaheurísticas de búsqueda, considerando tanto búsqueda local como global, y las estrategias evolutivas. En la cuarta sección se analiza y enumeran las principales características deseables de las mismas. El trabajo finaliza con unas breves conclusiones.

2. Las metaheurísticas

2.1. Concepto de metaheurística

La idea más genérica del término *heurístico* está relacionada con la tarea de resolver inteligentemente problemas reales usando el conocimiento disponible. El término heurística proviene de la palabra griega *eureka* cuyo significado está relacionado con el concepto de encontrar y se vincula a la supuesta exclamación de

Arquímedes al descubrir su famoso principio.

La concepción más común en IA es interpretar que heurístico es el *calificativo* apropiado para los procedimientos que, empleando conocimiento acerca de un problema y de las técnicas aplicables, tratan de aportar soluciones (o acercarse a ellas) usando una cantidad de recursos (generalmente tiempo) razonable. En un problema de optimización, aparte de las condiciones que deben cumplir las soluciones factibles del problema, se busca la que es óptima según algún criterio de comparación entre ellas. En Investigación Operativa, el término heurístico se aplica a un procedimiento de resolución de problemas de optimización con una concepción diferente. Se califica de *heurístico* a un procedimiento para el que se tiene un alto grado de confianza en que encuentra soluciones de alta calidad con un coste computacional razonable, aunque no se garantice su optimalidad o su factibilidad, e incluso, en algunos casos, no se llegue a establecer lo cerca que se está de dicha situación. Se usa el calificativo heurístico en contraposición a *exacto*, que se aplica los procedimientos a los que se les exige que la solución aportada sea óptima o factible. Una solución *heurística* de un problema es la proporcionada por un método heurístico, es decir, aquella solución sobre la que se tiene cierta confianza de que es factible y óptima, o de que alcanza un alto grado de optimalidad y/o factibilidad. También es usual aplicar el término heurística cuando, utilizando el conocimiento que se tiene del problema, se realizan modificaciones en el procedimiento de solución del mismo que, aunque no afectan a la complejidad del procedimiento, mejoran el rendimiento en su comportamiento práctico.

Unas heurísticas para resolver un problema de optimización pueden ser más *generales* o *específicas* que otras. Los métodos heurísticos específicos deben ser diseñados a propósito para cada problema, utilizando toda la información disponible y el análisis del modelo. Los procedimientos específicos bien diseñados suelen tener un rendimiento significativamente más alto que las heurísticas generales. Las heurísticas más generales, por el contrario, presentan otro tipo de ventajas, como la sencillez, adaptabilidad y robustez de los procedimientos. Sin embargo, las heurísticas generales emanadas de las metaheurísticas pueden mejorar su rendimiento utilizando recursos computacionales y estrategias inteligentes.

El término metaheurísticas se obtiene de anteponer a *heurística* el sufijo *meta* que significa “más allá” o “a un nivel superior”. Las concepciones actuales de lo que es una metaheurística están basados en las diferentes interpretaciones de lo que es una forma inteligente de resolver un problema. Las *metaheurísticas* son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento. El término metaheurística apareció por primera vez en el artículo seminal sobre búsqueda *tabú* de Fred Glover en 1986 [Glo86]. A partir de entonces han surgido multitud de propuestas de pautas para diseñar buenos procedimientos para resolver ciertos problemas que, al ampliar su campo de aplicación, han adoptado la denominación de metaheurísticas.

Las Hiperheurísticas es un concepto ligeramente diferente. Una hiperheurística se concibe como una herramienta o metaheurística de alto nivel que permite controlar adaptativamente varias heurísticas o metaheurísticas de bajo nivel de conocimiento del problema, de forma que aplicando sólo heurísticas fáciles de usar se pueden conseguir soluciones equiparables a las obtenidas por heurísticas de alto grado de conocimiento del problema. La diferencia fundamental entre una metaheurística y una hiperheurística es que ésta trata de elegir, en cada mo-

mento de decisión la metaheurística apropiada y no la solución a un problema; las hiperheurísticas sólo manipulan soluciones del problema indirectamente. Sin embargo, las hiperheurísticas se suelen diseñar a partir de una metaheurística. Entre ellas se encuentran muchas de las aplicaciones de los algoritmos genéticos en los que estos se usan para ajustar dinámicamente algunos parámetros se conciben como hiperheurísticas. La versión de la metaheurística VNS que selecciona de forma sistemática diferentes búsquedas locales descendientes es otro ejemplo.

La aparición más temprana de una hiperheurística es [FT61], es un término que se viene utilizando por Peter Cowling, Graham Kendall y Eric Soubeiga, de la Universidad de Nottingham, en diversos trabajos en las *Metaheuristic International Conference* [PS01], algunos working papers y en un artículo reciente [EBS].

2.2. Tipos de metaheurísticas

Las metaheurísticas son estrategias para diseñar procedimientos heurísticos. Por tanto, los tipos de metaheurísticas se establecen, en primer lugar, en función del tipo de procedimientos a los que hace referencia. Algunos de los tipos fundamentales son las metaheurísticas para los métodos de relajación, las metaheurísticas para los procesos constructivos, las metaheurísticas para las búsquedas por entornos y las metaheurísticas para los procedimientos evolutivos.

- Las metaheurísticas de *relajación* se refieren a procedimientos de resolución de problemas que utilizan relajaciones del modelo original (es decir, modificaciones del modelo que lo hacen más fácil de resolver), cuya solución facilita la solución del problema original.
- Las metaheurísticas *constructivas* se orientan a los procedimientos que tratan de la obtención de una solución a partir del análisis y selección paulatina de las componentes que la forman.
- Las metaheurísticas *de búsqueda* guían los procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explotar las estructuras de entornos asociadas.
- Las metaheurísticas *evolutivas* enfocadas a los procedimientos basados en conjuntos de soluciones que evolucionan sobre el espacio de soluciones.

Algunas metaheurísticas surgen combinando metaheurísticas de distinto tipo, como la metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) [RR03], [RG03], que combina una fase constructiva con una fase de búsqueda de mejora. Otras metaheurísticas se centran en el uso de algún tipo de recurso computacional o formal, especial como las redes neuronales, los sistemas de hormigas o la satisfacción de restricciones y no se incluyen claramente en ninguno de los cuatro tipos anteriores.

Por otro lado, de una u otra forma, todas las metaheurísticas se pueden *concebir* como estrategias aplicadas a procesos de búsqueda, donde todas las situaciones intermedias en el proceso de resolución del problema se interpretan como elementos de un espacio de búsqueda, que se van modificando a medida que se aplican las distintas operaciones diseñadas para llegar a la solución

definitiva. Por ello, y porque los procesos de búsqueda heurística constituyen el paradigma central de las metaheurísticas, es frecuente interpretar que el término metaheurística es aplicable esencialmente a los procedimientos de búsqueda sobre un espacio de soluciones alternativas. Por este mismo motivo se dedica una parte importante de este trabajo a las metaheurísticas de búsqueda.

2.2.1. Metaheurísticas de relajación

Una cuestión relevante al abordar un problema real es la obtención de un modelo que permita emplear una técnica de resolución apropiada. Si con este modelo el problema resulta difícil de resolver se acude a modelos modificados en los que es más sencillo encontrar buenas soluciones o en los que los procedimientos son más eficientes. Una relajación de un problema es un modelo simplificado obtenido al eliminar, debilitar o modificar restricciones (u objetivos) del problema real. En cualquier formulación siempre existe algún grado de simplificación, lo que puede afectar en mayor o menor medida al ajuste a la realidad de los procedimientos de resolución y de las soluciones del problema propuestas. Los modelos muy ajustados a la realidad suelen ser muy difíciles de resolver, y sus soluciones difíciles de implementar exactamente, por lo que se acude a modelos relajados. Las *metaheurísticas de relajación* son estrategias para el empleo de relajaciones del problema en el diseño de heurísticas. Se refieren al diseño, tanto de procedimientos que utilizan formulaciones relajadas del problema para proponer sus soluciones, como soluciones del problema, como de procedimientos que usan dichas relajaciones para guiar las operaciones realizadas para su resolución.

Muchas heurísticas de relajación modifican elementos del problema para proponer la solución de estas modificaciones como solución heurística del problema original. Las buenas relajaciones son las que simplifican el problema y hacen más eficientes los procedimientos de solución, pero cuya resolución proporciona muy buenas soluciones del problema original. Por ejemplo, para un problema de programación lineal entera, su relajación lineal consiste en ignorar la restricción de que las variables sean enteras. Esta se utiliza frecuentemente para aplicar procedimientos eficientes de programación lineal, como el método del Simplex, a dicha relajación y proponer una solución entera muy próxima a la solución del problema relajado.

Entre las metaheurísticas de relajación se encuentran los métodos de relajación lagrangiana [Bea93], [Gui02] o de restricciones subordinadas. Otras metaheurísticas de relajación alteran las restricciones o los objetivos del problema para usar su solución en la conducción de la búsqueda de la solución del problema original. Esta modificación puede estar encaminada a relajar las restricciones a las que debe estar sometida la solución, permitiendo que el recorrido bordee la región factible para acercarse al óptimo global incluso desde la región no factible. Otras estrategias modifican la función objetivo para obtener, de forma más rápida, valoraciones aproximadas (por exceso o por defecto) de la calidad de la solución que orientan la búsqueda, al menos en los estados iniciales. Es frecuente encontrar problemas en los que evaluar la función objetivo puede significar resolver otro problema de gran dificultad, realizar un proceso de simulación o realizar algún tipo de inversión o consumo de recursos. Para estos problemas es muy útil encontrar funciones sencillas de cálculo que den una idea aproximada de la calidad de las soluciones sin necesidad de una evaluación

ajustada de la función objetivo.

2.2.2. Metaheurísticas constructivas

Las heurísticas constructivas aportan soluciones del problema por medio de un procedimiento que incorpora iterativamente elementos a una estructura, inicialmente vacía, que representa a la solución. Las *metaheurísticas constructivas* establecen estrategias para seleccionar las componentes con las que se construye una buena solución del problema. Entre las metaheurísticas primitivas en este contexto se encuentra la popular estrategia voraz o *greedy*, que implica la elección que da mejores resultados inmediatos, sin tener en cuenta una perspectiva más amplia. Dentro de este tipo de metaheurística, destaca la aportación de la metaheurística GRASP [RR03], [RG03] que, en la primera de sus dos fases, incorpora a la estrategia *greedy* pasos aleatorios con criterios adaptativos para la selección de los elementos a incluir en la solución.

2.2.3. Metaheurísticas de búsqueda

El tipo de metaheurística más importante es el de las *metaheurísticas de búsqueda*, que establecen estrategias para recorrer el espacio de soluciones del problema transformando de forma iterativa la solución de partida.

La concepción primaria de heurística más frecuente era la de alguna regla inteligente para mejorar la solución de un problema que se aplicaba iterativamente mientras fuera posible obtener nuevas mejoras. Tales procesos se conocen como búsquedas monótonas (descendentes o ascendentes), algoritmos escaladores (*hill-climbing*) o búsquedas locales. Esta última denominación obedece a que la mejora se obtiene en base al análisis de soluciones similares a la que realiza la búsqueda; denominadas *soluciones vecinas*. Estrictamente hablando, una *búsqueda local* es la que basa su estrategia en el estudio de soluciones del vecindario o entorno de la solución que realiza el recorrido. Las metaheurísticas de búsqueda local son las estrategias o pautas generales para diseñar métodos de búsqueda local, como las que utilizan la estrategia voraz o *greedy*. Esta metaheurística establece como pauta, una vez consideradas cuales son las soluciones que intervienen en el análisis local, elegir iterativamente la mejor de tales soluciones mientras exista alguna mejora posible.

Se suele asumir que las búsquedas locales sólo modifican la solución que realiza el recorrido mediante una mejora en su propio entorno. El principal inconveniente de estas búsquedas locales es que se quedan atrapadas en un óptimo local, una solución que no puede ser mejorada por un análisis local. Por ello, el propósito fundamental de las primeras metaheurísticas era extender la búsqueda local para continuarla más allá de los óptimos locales, denominándose *Búsqueda Global*.

Las metaheurísticas de búsqueda global incorporan pautas para tres formas básicas de escapar de los óptimos locales de baja calidad: volver a iniciar la búsqueda desde otra solución de arranque, modificar la estructura de entornos que se está aplicando y permitir movimientos o transformaciones de la solución de búsqueda que no sean de mejora. Surgen así, respectivamente, las metaheurísticas de *arranque múltiple* [Mar03], [MM03] que establecen pautas para reiniciar de forma inteligente las búsquedas descendentes, las metaheurísticas de *entorno variable* que modifican de forma sistemática el tipo de movimiento con

el objeto de evitar que la búsqueda se quede atrapada por una estructura de entornos rígida y las metaheurísticas de *búsquedas no monótonas* que también aplican movimientos de no mejora durante el recorrido de búsqueda.

Las metaheurísticas para búsquedas no monótonas controlan los posibles movimientos de empeoramiento de la solución mediante criterios de aceptación estocásticos o utilizando la memoria del proceso de búsqueda. Las metaheurísticas de *búsqueda estocásticas* establecen pautas para regular la probabilidad de aceptar transformaciones que no mejoren la solución. El *Recocido Simulado* [KGV83], [DD03] es el exponente más importante de este tipo de metaheurísticas donde la probabilidad de aceptación es una función exponencial del empeoramiento producido. Las metaheurísticas de *búsqueda con memoria* utilizan información sobre el recorrido realizado para evitar que la búsqueda se concentre en una misma zona del espacio. Fundamentalmente se trata de la *Búsqueda Tabú* [GL97], [GM03] cuya propuesta original prohíbe temporalmente soluciones muy parecidas a las últimas soluciones del recorrido.

2.2.4. Metaheurísticas evolutivas

Las *metaheurísticas evolutivas* establecen estrategias para conducir la evolución en el espacio de búsqueda de conjuntos de soluciones (usualmente llamados *poblaciones*) con la intención de acercarse a la solución óptima con sus elementos. Solo se distinguen de las metaheurísticas de búsqueda en que es un conjunto de soluciones y no una solución el que evoluciona sobre el espacio de búsqueda, por esta razón también se tratan en el apartado siguiente de metaheurísticas de búsqueda. El aspecto fundamental de las heurísticas evolutivas consiste en la interacción entre los miembros de la población frente a la búsqueda que se guían por la información de soluciones individuales.

Las diferentes metaheurísticas evolutivas se distinguen por la forma en que combinan la información proporcionada por los elementos de la población para hacerla evolucionar mediante la obtención de nuevas soluciones. Los algoritmos genéticos [Gol89], [Ree03] y meméticos [PM03], [MCP03], [Mos99] y los de estimación de distribuciones [LL02] [LLM03] emplean fundamentalmente procedimientos aleatorios, mientras que las metaheurísticas de búsqueda dispersa, de re-encadenamiento de caminos (*Path Relinking*) [LGM03], [ML03] o los algoritmos biónicos emplean procedimientos sistemáticos.

2.2.5. Otros tipos de metaheurísticas

Otras metaheurísticas que aparecen en varias clasificaciones corresponden a tipos intermedios entre los anteriores [SVVW80], [ZEV89]. Entre ellas destacan las metaheurísticas de descomposición y las de memoria a largo plazo.

Las *metaheurísticas de descomposición* establecen pautas para resolver un problema determinando subproblemas a partir de los que se construye una solución del problema original. Se trata de metaheurísticas intermedias entre las de relajación y las constructivas, ya que se refieren básicamente a las características que se pretenden obtener en los subproblemas y a cómo integrar las soluciones de estos subproblemas en una solución del problema original. El objetivo fundamental es obtener subproblemas significativamente más fáciles de resolver que los originales, y cuyas soluciones puedan ser utilizadas efectivamente. Este es el tipo de metaheurística más apropiada para la aplicación de estrategias de

paralelización, donde es muy importante el equilibrio entre los subproblemas obtenidos.

Las *metaheurísticas de memoria a largo plazo* constituyen el caso más relevante de las metaheurísticas de aprendizaje y se sitúan entre las de arranque múltiple y las derivadas de la búsqueda tabú. Por ejemplo, diversas metaheurísticas se refieren al uso de información sobre las características y propiedades comunes a soluciones de alta calidad o sobre las decisiones de mejora adoptadas durante el proceso de solución. Esta información permite mejorar el rendimiento de la búsqueda de arranque múltiple ajustando los parámetros que modulan la exploración y la explotación del proceso. Se incluyen en las metaheurísticas de *aprendizaje* ya que son capaces de emplear información obtenida en la aplicación del propio procedimiento, tanto a un problema específico como a un tipo o clase específica de problemas.

2.3. Relevancia de las metaheurísticas

La relevancia de las metaheurísticas se refleja en la publicación de libros dedicado expresamente a metaheurísticas, la aparición de artículos, especialmente revisiones o reviews, números especiales de revistas, recursos disponibles en la red (páginas webs), programas o software específico, etc.

2.3.1. Libros sobre metaheurísticas

La relevancia de las metaheurísticas se refleja en la publicación de libros sobre este campo en los últimos años. Hasta 1995 tratando diferentes metaheurísticas aparece el texto de C. Reeves *Modern Heuristic Techniques for Combinatorial Problems* [Ree93] que en 1993 recoge capítulos dedicados algunas de las metaheurísticas más relevantes y el libro editado por V.J. Rayward-Smith *Applications of Modern Heuristic Methods* [RS95] en 1995 que recoge aplicaciones de diferentes metaheurísticas. Además aparecen los siguientes libros dedicados a una metaheurística en particular [Ack87], [AK89], [Gol89], [GLTW93] y [GW93]. En 1996 aparece el texto en español coordinado por Adenso Díaz y otros titulado *Optimización Heurística y Redes Neuronales* [DGG⁺96] con capítulo dedicados a las metaheurísticas mas importantes. Además antes de 2000 aparecen varios textos dedicadas a las diversas metaheurísticas [RSORS96], [OK96], [LO96], [CDG99] y [VMOR99] y pocos textos a un tipo más particular de metaheurísticas [AL96], [GL97] y [BDT99]. Más recientemente, desde 2000 se han publicado, varios textos, destacando el excelente caracter pedagógico del texto de Michalewicz y Fogel *How to Solve It: Modern Heuristics* [MF00]. Además aparecen los textos dedicados a aplicaciones en telecomunicaciones [OCS00] y [BY02], a implementaciones [VW02] o dedicadas en en particular al Scatter Search [LM02b] o a los algoritmos genéticos [RR02], además de las collecciones [LG00] y [RH01] derivados de congresos respectivos. Finalmente el texto más reciente es el de Glover y Kochenberger [GK03] aunque ya se anuncian nuevos títulos.

2.3.2. Artículos sobre metaheurísticas

La literatura en español es escasa, además del texto ya mencionado coordinado por A. Díaz [DGG⁺96] y el de caracter docente [VP00] aparecen pocos

artículos de revisión [BMM03] y la monografía ([MM03]) de la revista *Inteligencia Artificial* disponible en la red¹.

Diversos volúmenes especiales sobre metaheurísticas han venido apareciendo en diversas colecciones editoriales o revistas periódicas de los campos de Investigación Operativa, Inteligencia Artificial, Ingeniería y Ciencias de la Computación.

Además, en estas publicaciones se observa un incremento considerable del número de trabajos que incluyen procedimientos heurísticos en los que se realizan planteamientos estándares de las metaheurísticas.

Desde 1985 se viene publicando la revista *Journal of Heuristics*², que concentra una parte importante de las publicaciones en este campo

Otras revistas relevantes en metaheurísticas son:

- *Computational Optimization and Applications* ³.
- *Computers and Operations Research* ⁴.
- *European Journal of Operational Research* ⁵.
- *INFORMS Journal on Computing* ⁶.
- *Journal of Combinatorial Optimization* ⁷.
- *Evolutionary Computation* ⁸.

2.3.3. Enlaces relacionados

Diversas asociaciones, entidades y proyectos ofrecen información sobre metaheurísticas en Internet.

La red *HEUR* ⁹ se creó en 2002 financiada por el Ministerio de Ciencia y Tecnología a través del proyecto TIC2002-10886E con 15 grupos y 101 miembros. Con el objetivo de la red es la difusión de los métodos metaheurísticos, así como la interacción entre los distintos grupos, la red proporciona en este portal tutoriales y artículos de investigación sobre procedimientos metaheurísticos.

El grupo *EURO Working Group/European Chapter* en Metaheurísticas (EU/ME)¹⁰ proporciona abundante información en su página sobre las distintas metaheurísticas. La red de excelencia sobre Metaheurística (*Metaheuristics Network* ¹¹ financiado por la unión europea (HPRN-CT-1999-00106) dedica especial atención a algunas metaheurísticas.

Sobre metaheurísticas específicas se obtiene información en los siguientes enlaces:

¹<http://sensei.ieec.uned.es/cgi-bin/aepia/contenidoNum.pl?numero=19>

²<http://www.kluweronline.com/issn/1381-1231>

³<http://www.kluweronline.com/issn/0926-6003/>

⁴<http://www.elsevier.nl/inca/publications/store/3/0/0/>

⁵<http://www.elsevier.nl/homepage/sae/orms/eor/menu.sht>

⁶<http://joc.pubs.informs.org/>

⁷<http://www.kluweronline.com/issn/1382-6905/>

⁸<http://mitpress.mit.edu/catalog/item/default.asp?tttype=4&tid=25>

⁹<http://heur.uv.es/>

¹⁰<http://143.129.203.3/eume/welcome.htm>

¹¹<http://www.metaheuristics.org/>

- El sitio oficial de la Búsqueda por Entorno Variable (VNS) ¹².
- La página sobre Ant Colony Optimization de M. Dorigo ¹³.
- La página sobre Algoritmos Meméticos de Moscato ¹⁴.
- La página Reactive Search de R. Battiti ¹⁵.
- La página sobre Tabu Search de C. Rego and F. Glover) ¹⁶.
- Sobre Simulated Annealing ¹⁷.
- La red EvoNET¹⁸ sobre computación evolutiva.

Además la lista *Modern heuristics mailing list* distribuye información relevante sobre metaheurísticas ¹⁹.

Los congresos más relevantes en este campos son los siguientes:

- Los congresos españoles de Metaheurísticas, Algoritmos Evolutivos y Bioinspirado que en su cuarta edición (MAEB'04) se ha celebrado en Córdoba ²⁰.
- Las *European Conference on Evolutionary Computational* que en su cuarta edición EVOCOP 2004²¹
- las *Metaheuristics International Conference* cuya quinta edición se celebró en Kyoto (Japón) (MIC 2003) y la sexta se celebrará en el 2005 en Viena ²²

2.3.4. Software sobre metahurísticas

Otro aspecto que refleja la relevancia que alcanzan las metaheurísticas es una panorámica de los avances y propuestas exitentes en entornos de desarrollo software de metaheurísticas.

Se han ido realizando diversos intentos de desarrollo de sistemas, entornos y herramientas software para la aplicación de metaheurísticas en la resolución de problemas de optimización combinatoria. Inicialmente la mayoría de los proyectos e investigaciones en este sentido, se llevan a cabo en relación al desarrollo de programas y código eficiente donde se aplican metaheurísticas específicas a problemas concretos. La idea de desarrollar entornos generales chocaba con la posibilidad de su ineficiencia, ligada tanto a la no adaptación al problema como a la dificultad de encontrar modelos generales de representación de los mismos.

Ambas cuestiones son puntos críticos en el desarrollo de herramientas, cuya misión se encuentra en la reducción del tiempo de especificación de problemas y

¹²<http://vnsheuristic.ull.es>

¹³<http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>

¹⁴http://www.densis.fee.unicamp.br/~moscato/memetic_home.html

¹⁵<http://rtm.science.unitn.it/~battiti/reactive.html>

¹⁶<http://www.tabusearch.net/>

¹⁷<http://www.taygeta.com/annealing/simanneal.html>

¹⁸<http://evonet.dcs.napier.ac.uk/>

¹⁹<http://www.jiscmail.ac.uk/lists/modern-heuristics.html>

²⁰<http://maeb04.ayrna.org/indexf.htm>

²¹<http://evonet.dcs.napier.ac.uk/eurogp2004/>

²²<http://www-or.amp.i.kyoto-u.ac.jp/mic2003/>

la solución del mismo. Desde el punto de vista de la especificación del problema, la condición de que estos entornos sean de propósito general implica el uso de estructuras de datos adecuadas para el almacenamiento de información sobre un amplio rango de problemas. La cuestión es si existe una representación adecuada y general para diversos tipos de problemas o en general esta representación va a condicionar la eficiencia de los procedimientos de resolución. Así mismo es necesario contar con representaciones que tienen que ser lo suficientemente adaptables para especificar el problema de optimización, sus variables, restricciones y funciones objetivo y añadir cambios a los mismos. Por tanto disponer de herramientas de modelización de problemas que permitan desarrollar modelos y prototipos. Desde el punto de vista de las metaheurísticas, estas herramientas deben estar capacitadas para especificar e implementar diversas estrategias de resolución, implementar diversas metaheurísticas generales que se puedan adaptar a problemas específicos y que incluya una interfaz adecuada a la posibilidad de parametrizar los distintos criterios que intervienen en la estrategia, como operadores de búsqueda, definición de entornos, criterios de parada, etc. Además es de interés que contenga los propios resolutores de problemas que a partir de la representación permita su ejecución y la obtención de soluciones sobre un rango amplio de problemas.

El camino para abordar esta cuestión ha pasado inicialmente por la producción de código general o esqueletos generales de procedimientos metaheurísticos bajo diversos paradigmas procedimentales, orientados a objetos o funcionales y sobre problemas específicos [Con92], [Osm93].

La mayoría de los avances en entornos de desarrollo de resolución de problemas de optimización han estado basados en el paradigma orientado a objetos, ya que ha permitido la estructuración simple del código, el incremento de la modularidad y la reusabilidad. Esto ha hecho posible el desarrollo de librerías de componentes, que se pueden embeber en otros programas o entornos basados en componentes. La librerías están definidas como un conjunto de clases diseñadas mediante patrones que sirven para resolver un conjunto de problemas similares.

Los primeros resultados se obtienen en proyectos del mismo tipo de los anteriores en las que se aplican metaheurísticas concretas a problemas específicos [FHL96] o también algo más avanzados el desarrollo de una librería de clases para metaheurísticas generales como algoritmos genéticos, búsqueda tabú o recorrido simulado o estrategias híbridas de ellas, descritas en [Woo97].

Posteriormente se han desarrollados varios entornos genéricos entre los que se encuentra la arquitectura desarrollada utilizando patrones de diseño y basados en diversas estrategias de búsqueda local. Arquitectura permiten la comparación de estrategias e incluyen clases que facilitan la construcción inicial de soluciones, la generación de entornos y la selección de diversos criterios para los movimientos [ACR98], o también el entorno de desarrollo usando diversos templates como librerías de componentes reusables [FVW98], [FVW98], [Vos01], aproximación que incluye *HotFrame* (*Heuristic Optimization Framework*) que proporciona componentes adaptables en C++, incluyendo diversas metaheurísticas y una arquitectura de colaboración entre las distintas componentes y clases específicas de aplicaciones y permitiendo la hibridación y la incorporación de nuevas metaheurísticas.

Un entorno similar orientado a objetos que puede ser usado como herramienta general para desarrollar e implementar metaheurísticas de búsqueda local en

C++ es *LOCAL++* [SLC99], compuesta por una jerarquía de clases conteniendo templates de metaheurísticas de búsqueda, y que permiten especialización para abordar problemas específicos, combinaciones entre ellas o también la creación de nuevas estrategias, así mismo facilita al diseñador la creación de esquemas conceptuales de la aplicación. Otros entornos similares pero descritos en Java, es *TabOOBuilder* [GP99] o el entorno *MAFRA* [KS00] entorno orientado a objetos especializado en algoritmos meméticos.

Utilizando los mismos paradigmas, tanto en lo que respecta a la especificación y resolución de problemas mediante metaheurísticas de búsqueda local, como respecto al paradigma de desarrollo orientado a objetos, se han desarrollado diversas herramientas de desarrollo. Estas herramientas están influenciadas, en parte, por avances en los lenguajes de especificación matemática *AMPL* [FGK93], el uso de librerías estándar de resolución de problemas como *CPLEX* o de manera similar a los desarrollos basados en programación por restricciones *OPL* [Hen99] o *Salsa* [LC98]. Todos ellos con la misma finalidad de resolver problemas de optimización, mediante entornos que faciliten la modelización de tales problemas, la especificación de los procedimientos de resolución y su resolución. En este marco nos encontramos con entornos de desarrollo potentes como *Localizer* [MH97], [MH99], un lenguaje de modelización de alto nivel de problemas que permite describir metaheurísticas de búsqueda de manera natural, reduciendo el tiempo de desarrollo con un alto grado de eficiencia, *Localizer++* [MH01], el conjunto de librerías abiertas que pueden extender las facilidades del lenguaje de modelización. Una herramienta orientada a objetos similar es *EasyLocal++* [GS01], cuyo predecesor es *LOCAL++*, para el desarrollo de metaheurísticas de búsqueda local compuesta de clases, que implementa partes invariantes de las estrategias y son especializadas por clases concretas con la parte dependiente del problema específico.

En paralelo a los anteriores, se han desarrollado resolutores u optimizadores de propósito general basados en metaheurísticas [RA00]. Otros en cambio sirven para aplicarlos a la resolución de problemas particulares de optimización, algunos basados en metaheurísticas específicas [Lag97] como *Evolver*, en algoritmos genéticos y *OptQuest* [LM02a] en búsqueda dispersa, son comerciales, donde además de capacidades de optimización tienen capacidades de simulación. Una versión no comercial de optimizador es *Genocop* que implementa algoritmos genéticos.

Un enfoque diferente pero con el mismo objetivo es la arquitectura multiagente desarrollada por Andrea Roli y Micherla Milano, donde introduce la tecnología de agentes inteligentes para el desarrollo de una arquitectura cooperativa de aplicación y desarrollo de metaheurísticas *MAGMA* [RM02].

Es significativo que el desarrollo de metaheurísticas para la resolución de problemas multiobjetivos ha tenido su réplica en el desarrollo de entornos genéricos como *METHOD* [CS01], entorno orientado a objetos desarrollado en C++, que proporciona soporte para la búsqueda local multiobjetivo.

3. Metaheurísticas de búsqueda

Las *metaheurísticas de búsqueda* aportan estrategias para afrontar la resolución de un problema realizando una búsqueda sobre un espacio cuyos elementos representan las soluciones candidatas alternativas. La representación de las solu-

ciones se realiza a través de una codificación que incluya toda la información necesaria para su identificación y evaluación. Una búsqueda sobre un espacio consiste en generar una sucesión de puntos del espacio pasando de uno a otro por medio de una serie de transformaciones o movimientos. Un procedimiento de búsqueda para resolver un problema de optimización realiza recorridos sobre el espacio de las soluciones alternativas y selecciona la mejor solución encontrada en el recorrido. Las metaheurísticas de búsqueda proporcionan pautas para obtener recorridos que, con alto rendimiento, proporcionen soluciones de alta calidad.

La descripción general del proceso de resolución de un problema es, partiendo de una situación inicial, aplicar iterativamente una operación para modificar la situación actual, hasta que se alcance la situación buscada. Un proceso de búsqueda basado en transformaciones o movimientos sobre un espacio de soluciones posibles consiste en la selección iterativa de movimientos para transformar una solución hasta que se cumpla cierto criterio de parada. El criterio de parada determina cuándo se considera resuelto el problema sin que sea necesario disponer, en una situación intermedia, de información de lo cerca que se está de solucionarlo. Sin embargo, las búsquedas inteligentes deben utilizar este y otro tipo de información tanto en el criterio de parada, como en la selección de los movimientos.

En los problemas de optimización, la selección de movimientos y el criterio de parada se diseñan teniendo en cuenta, al menos, un indicador de la calidad de las soluciones encontradas en el recorrido. La evaluación de la calidad de las soluciones se realiza a través de una o varias funciones objetivo, teniendo en cuenta las restricciones del problema. La estrategia de búsqueda establece los criterios y mecanismos que guiarán el recorrido. La estrategia de búsqueda puede incorporar técnicas de una o varias metaheurísticas, junto a heurísticas específicas para el problema. Por su generalidad, la descripción y análisis de las metaheurísticas de búsqueda se realizan sobre problemas de optimización.

Un *problema de optimización* es aquel cuya solución implica encontrar en un conjunto de soluciones candidatas alternativas aquella que mejor satisface unos objetivos. Los problemas de optimización surgen en muchísimos campos científicos y su solución es de crucial importancia para el éxito de multitud de tareas de Inteligencia Artificial. Cada problema de optimización se especifica estableciendo cuál es el espacio de soluciones alternativas y los objetivos perseguidos. Los objetivos se formalizan por una o varias funciones que hay que maximizar o minimizar (suponemos, en la descripción de los métodos de solución, que se trata de minimizar). Formalmente, el problema de optimización con una función objetivo se compone del espacio de soluciones S y la función objetivo f . Resolver el problema de optimización (S, f) consiste en determinar una solución óptima, es decir, una solución factible $x^* \in S$ tal que $f(x^*) \leq f(x)$, para cualquier $x \in S$.

El espacio de soluciones alternativas se pueden expresar por la asignación de valores a algún conjunto finito de variables $X = \{X_i : i = 1, 2, \dots, n\}$. Si por U_i se denota al dominio o universo (conjunto de los valores posibles) de cada una de estas n variables, una solución del problema se obtiene de seleccionar los valores x_i asignados a cada una de las variable X_i del dominio U_i que, sometido a ciertas restricciones, optimiza una función objetivo f . El universo de soluciones se identifica con el conjunto $U = \{x = (x_i : i = 1, 2, \dots, n) : x_i \in U_i\}$. Las restricciones del problema reducen el universo de soluciones a un subconjunto

de soluciones $S \subseteq U$, denominado espacio factible.

3.1. Estructura de entornos

Los procedimientos de *búsqueda por entornos* recorren el espacio de soluciones U mediante un conjunto de transformaciones o movimientos. Las soluciones que se obtienen de otra mediante uno de los movimientos posibles se denominan vecinas de ésta y constituyen su *entorno*. El conjunto de movimientos posibles da lugar a una relación de vecindad y una *estructura de entornos* en el espacio de soluciones, su elección es trascendental e interviene en ella, la implementación y evaluación eficiente de los movimientos y las propiedades de la estructura de entorno resultante. El esquema general de un procedimiento de búsqueda por entornos consiste en generar una solución inicial y, hasta que se cumpla el criterio de parada, seleccionar iterativamente un movimiento para modificar la solución. Las soluciones son evaluadas mientras se recorren y se propone la mejor solución del problema encontrada.

El entorno de una solución está constituido por las soluciones a las que se puede acceder desde ella por uno de los movimientos posibles. Formalmente, una *estructura de entornos* sobre un espacio o universo de búsqueda U es una función $E : U \rightarrow 2^U$ que asocia a cada solución $x \in U$ un entorno $E(x) \subseteq U$ de soluciones vecinas a x . Gran cantidad de métodos heurísticos propuestos en la literatura pertenece a la clase de procedimientos de búsqueda por entornos [MF00], [Pir94].

La descripción en pseudocódigo de la búsqueda por entornos se muestra en la figura 1.

```
1: Procedure Búsqueda por Entornos
2: {
3:    $x \leftarrow \text{GenSol}(U)$ ;
4:    $x^* \leftarrow x$ ;
5:   do {
6:      $x \leftarrow \text{SelSol}(E(x))$ ;
7:     if ( $\text{Obj}(x) \prec \text{Obj}(x^*)$ )
8:        $x^* \leftarrow x$ ;
9:   }while (no CriPar)
10: }
```

Figura 1: Búsqueda por Entornos

La elección de la estructura de entornos es fundamental en el éxito de los procesos de búsqueda ya que determina la calidad del conjunto de movimientos aplicados. Aparte de la factibilidad y el grado de mejora de los movimientos aplicados es importante la versatilidad de los mismos.

Los movimientos combinados aparecen al ejecutar sucesivamente varios movimientos sobre una solución. Una adecuada combinación de movimientos enriquece los entornos, con lo que se pueden realizar pasos más amplios en el acercamiento al óptimo, aunque se corre el riesgo de perjudicar la eficiencia del algoritmo al tener que contemplar un número mayor de movimientos posibles en el proceso de selección.

Otra característica importante de los movimientos es la factibilidad de las soluciones aportadas. Los movimientos factibles son aquellos que siempre proporcionan una solución factible. Esto puede estar ligado o no al hecho de que se aplique sólo a soluciones factibles. En muchos casos, aplicar movimientos más simples, pero no necesariamente factibles, y descartar las soluciones producidas que no sean factibles, es menos eficiente que adaptar el diseño de los movimientos para que sean factibles, sobre todo cuando dicha comprobación es costosa o cuando la probabilidad de que resulte factible sea baja. Formalmente, los procedimientos que sólo consideran movimientos factibles están asociados al concepto, algo más restrictivo, de estructura de entornos como una función $E : S \rightarrow 2^S$ que asocia a cada solución factible $x \in S$ un entorno $E(x) \subset S$ de soluciones factibles vecinas a x .

Las principales metaheurísticas de búsqueda por entornos que se describen más adelante se centran sólo en el procedimiento de selección del movimiento. Sin embargo, además de la selección de la propia estructura de entornos sobre la que articular la búsqueda, existen otras cuestiones relevantes en el éxito del procedimiento de búsqueda por entornos, como son: la evaluación de la función objetivo, el procedimiento de generación de la solución inicial y el criterio de parada.

La posibilidad de realizar una evaluación eficiente de la solución obtenida tras el movimiento es especialmente importante en aquellos problemas en los que la evaluación de la función objetivo es costosa. Son aplicables las pautas de las metaheurísticas de relajación para evitar cálculos excesivos en la obtención de valoraciones exactas que no son imprescindibles en la conducción de la búsqueda. Además, se puede contar con procedimientos que evalúan la calidad de los movimientos sin tener que realizar una evaluación completa de la nueva solución desde cero. Para ello se utilizan procedimientos que actualizan inmediatamente el valor de la función objetivo tras el movimiento, utilizando el valor anterior y los cambios producidos por el movimiento.

Las pautas de las metaheurísticas constructivas se utilizan para el diseño de los procedimientos de generación de la solución inicial. En este sentido, las características fundamentales son la calidad y dispersión de la solución de arranque desde la que se inician las búsquedas. La metaheurística GRASP propone un procedimiento para conseguir diferentes soluciones de alta calidad.

Por último, en relación a los criterios de parada, los parámetros más corrientes se refieren a un límite al número de iteraciones, movimientos, operaciones elementales o tiempo de cómputo total o sin que se produzca alguna mejora.

Otras dos características fundamentales en el procedimiento de búsqueda por entorno resultante de aplicar metaheurísticas son las capacidades de exploración y de explotación. La *exploración* se refiere a la capacidad del método para explorar las diferentes regiones del espacio de búsqueda para alcanzar la zona en la que se encuentra la solución del problema. La *explotación* de la búsqueda se refleja en el esfuerzo y capacidad por mejorar las soluciones con las que trabaja el procedimiento. Existe un amplio consenso en que estas dos características deben modularse adecuadamente para conseguir el éxito práctico de las aplicaciones de las metaheurísticas.

3.2. Búsquedas locales

El término *local* se emplea con bastante frecuencia en los estudios teóricos y prácticos del campo de las metaheurísticas de búsqueda. Las estructuras de entorno suelen reflejar algún concepto de proximidad o vecindad entre las soluciones alternativas del problema. Por tanto, el análisis del entorno de la solución actual en el recorrido de búsqueda para decidir cómo continuarla, representa un estudio local del espacio de búsqueda. Las *metaheurísticas de búsqueda local* establecen pautas de selección de la solución del entorno a la solución actual, dando lugar a búsquedas locales heurísticas con alto rendimiento. Las búsquedas locales no informadas sólo tienen en cuenta la estructura de entornos para guiar la búsqueda. Las búsquedas monótonas utilizan la evaluación de la función objetivo para admitir sólo cambios en la solución actual que supongan una mejora. Por tanto, las búsquedas locales monótonas quedan atrapadas al llegar a una solución que no admite mejora dentro de su entorno. Las búsquedas globales emplean diversos métodos para escapar de esta situación. A continuación analizamos los aspectos más relevantes de las metaheurísticas para estos procedimientos.

3.2.1. Búsquedas no informadas

Las estrategias de búsqueda por entornos no informadas son aquellas búsquedas locales que sólo prestan atención a la estructura de entornos en el espacio de búsqueda y no utilizan información acerca del valor de la función objetivo en las soluciones encontradas. Las metaheurísticas de búsqueda no informadas aportan estrategias para organizar la exploración eficiente del espacio de búsqueda. Cuando estas pautas se aplican a la exploración del entorno en las búsquedas locales se traducen en metaheurísticas de búsqueda por entornos no informadas. Las metaheurísticas de búsqueda por entornos exhaustiva, parcial y aleatoria son las metaheurísticas de búsqueda no informadas más usuales.

Un recorrido exhaustivo de un espacio de búsqueda es el que incluye todos y cada uno de los elementos del espacio. Si el espacio de búsqueda es finito y no excesivamente grande, un procedimiento rudimentario para resolver el problema consiste en implementar un recorrido exhaustivo hasta encontrar la solución. En un problema de optimización, la búsqueda exhaustiva consiste en realizar un recorrido exhaustivo del espacio de soluciones del problema y tomar la mejor de ellas. Un recorrido exhaustivo del espacio se consigue empleando una ordenación (implícita o explícita) de todas las soluciones del espacio y utilizando una transformación que obtenga en cada iteración la solución siguiente en dicha ordenación. El procedimiento de generación de la solución inicial debe proporcionar la primera solución de dicha ordenación y el criterio de parada detectar cuándo se ha completado todo el espacio de búsqueda. La ordenación puede comprender sólo las soluciones factibles o un conjunto que las contenga. En este caso sólo habrá que considerar las soluciones factibles para elegir la mejor. A partir de la representación de las soluciones del espacio se determina la ordenación natural consistente en ir modificando sucesivamente los elementos que componen la solución. Dada una estructura de entornos para un problema, la búsqueda por entornos exhaustiva recorrerá sucesivamente y de forma exhaustiva los entornos de las soluciones visitadas. Si la estructura de entornos enlaza todas las soluciones del espacio, la búsqueda será exhaustiva, pero será necesario

evitar o controlar las repeticiones para impedir que se cicle indefinidamente.

En algunas circunstancias puede ser suficiente examinar sólo una parte del espacio de búsqueda para obtener una visión global de todo el espacio. Las metaheurísticas de *búsqueda parcial* establecen las pautas para organizar la selección de las soluciones a examinar. Para un problema de optimización, la búsqueda parcial aportará la mejor entre las soluciones examinadas como propuesta de solución. Si las soluciones a examinar se seleccionan de forma completamente al azar se trata de una búsqueda parcial aleatoria pura, conocida como método de *Monte Carlo*. La búsqueda parcial por entornos aleatoria aplica un método parcial para analizar el entorno de las soluciones del recorrido.

La metaheurística de *búsqueda por entornos aleatoria* consiste en seleccionar iterativamente al azar una solución del entorno de la solución actual. Se trata de un recorrido aleatorio puro o uniforme si la distribución de probabilidad en el entorno de la solución actual es uniforme o equiprobable. Para implementar esta metaheurística sólo es necesario disponer de un buen procedimiento que seleccione una solución vecina de la región factible, y una forma rápida de evaluar la nueva solución. La búsqueda se intensifica si la solución del entorno se selecciona de entre varias soluciones vecinas generadas al azar. La explotación de la búsqueda se ve incluso aumentada si el método de selección de las soluciones a examinar favorece a las de mayor calidad, o a las que se presume que lo van a ser, denominadas soluciones prometedoras. Por otro lado, si la selección parcial de las soluciones a examinar se realiza de forma que se evite la repetición de soluciones examinadas, se obtendrá un mejor aprovechamiento del tiempo de cómputo. La búsqueda parcial sistemática persigue evitar estas repeticiones manteniendo un alto grado de aleatoriedad.

Una búsqueda parcial sistemática se obtiene de un recorrido exhaustivo deteniendo la búsqueda sin necesidad de llegar a completar todo el espacio de soluciones. Si la parte del espacio recorrido es pequeña y las soluciones consecutivas, en la ordenación del espacio utilizada, son similares, la visión parcial del espacio de búsqueda sería demasiado sesgada. Para evitar este inconveniente se realiza la búsqueda parcial mediante un recorrido sistemático con arranque aleatorio. Esta estrategia consiste en determinar al azar una solución de arranque y una amplitud de paso no unitario para el recorrido. Además la ordenación es interpretada de forma cíclica (la siguiente de la última solución es la primera) para que el recorrido no se detenga al llegar al final de la ordenación. El recorrido sistemático de m elementos en un conjunto ordenado de n elementos se obtiene fijando una posición de arranque r y una amplitud de paso t . Conviene elegir la amplitud de paso t de forma que $m \cdot t > n$ y tal que t y n sean números primos entre sí o, al menos, con un mínimo común múltiplo suficientemente alto. Dada una estructura de entornos para un problema, la búsqueda por entornos parcial recorrerá sucesivamente y de forma parcial los entornos de las soluciones visitadas. El número de soluciones visitadas en el recorrido de cada entorno determina la intensidad de la búsqueda cuya regulación puede ser estática o dinámica.

3.2.2. Búsquedas locales monótonas

Las metaheurísticas de búsqueda anteriores no utilizan la información proporcionada por la evaluación de la función objetivo en la conducción de la búsqueda. Las estrategias de búsqueda pueden incorporar esta información al

método de búsqueda para guiar los movimientos aplicados. Las búsquedas informadas son aquellas que, explícita o implícitamente, utilizan información de la evaluación de la función objetivo. Las búsquedas locales (o por entornos) informadas son las que utilizan información de la función objetivo sólo en el entorno de la solución actual.

Las búsquedas monótonas sólo aceptan mejoras de la solución que realizan el recorrido. Las búsquedas locales monótonas son las búsquedas locales que sólo aplican movimientos que mejoren la solución actual del recorrido [AL96], [Pir94], [Pir96], [Gro92]. Frecuentemente se interpreta que las búsquedas locales persiguen siempre una mejora en los alrededores de la solución actual, aunque el término *local* hace referencia sólo a que se realiza un análisis en el entorno de la solución actual para guiar la búsqueda. Las búsquedas monótonas no estrictas aceptan también nuevas soluciones que igualan a la solución actual. Estas estrategias presentan la ventaja de que pueden escaparse de las mesetas o zonas llanas del espacio de búsqueda, pero tienen el inconveniente de que podría ciclarse indefinidamente dentro de una de tales mesetas.

La metaheurística básica de búsqueda por entorno monótona aleatoria consiste en seleccionar iterativamente una solución al azar del entorno de la solución actual, que es sustituida por ésta si se produce una mejora. La solución de partida se puede obtener por cualquier procedimiento arbitrario y el criterio de parada reflejará el estancamiento de la búsqueda en un mínimo local presumible cuando en un cierto número de intentos no se pueda mejorar la solución actual. Las metaheurísticas intensifican la búsqueda en torno a cada solución actual seleccionando la mejor entre una serie de soluciones del entorno, obtenidas por un procedimiento del mismo tipo. La intensidad de la búsqueda viene dada por el número o la proporción de soluciones vecinas de la solución actual entre las que se toma la mejor.

La metaheurística de intensificación oscilante consiste en hacer oscilar sistemáticamente entre dos valores extremos la intensidad de la búsqueda. La metaheurística de intensificación oscilante dinámica regula dinámicamente la intensidad de la búsqueda para intensificarla, hasta hacerla exhaustiva al acercarse al óptimo local, pero sin necesidad de encontrar la mejor solución vecina al comenzar los descensos. Una estrategia autónoma para esta regulación dinámica es, por ejemplo, aumentarla cada vez que no se mejore la solución, hasta alcanzar el tamaño del entorno, y disminuirla mientras se produzcan esas mejoras, sin llegar a anularla.

Las metaheurísticas de búsqueda sistemática mejoran el poder de exploración en el entorno de la solución actual, haciendo que las soluciones vecinas entre las que se selecciona la mejor sean distintas. Los procedimientos de búsqueda obtienen una ventaja con esta estrategia si las modificaciones necesarias para garantizar que las soluciones vecinas evaluadas sean distintas no hacen computacionalmente más costoso el procedimiento. El procedimiento se puede implementar, por ejemplo, asumiendo la ordenación implícita del entorno de cada solución y aplicando un procedimiento de muestreo sistemático con arranque aleatorio. Esta ordenación puede venir dada de forma natural o se puede derivar del procedimiento exhaustivo.

Las metaheurísticas de búsqueda local exhaustiva maximizan el poder de explotación de la búsqueda local al examinar, si es necesario, todo el entorno de la solución actual. Las metaheurísticas *voraz* y *ansiosa* aparecen al aplicar las dos reglas fundamentales de selección de esta solución. La metaheurística *voraz*

o (*Greedy*) con la regla de selección de *el mejor primero* y la metaheurística ansiosa (*Anxious*) con la regla de selección de *el primero mejor*. En la primera de ellas se selecciona siempre la mejor solución del entorno de la solución actual y en la segunda se selecciona la primera solución del entorno que mejore la solución actual. En la metaheurística por entornos voraz se recorren siempre todas las soluciones del entorno para seleccionar la mejor, mientras que en la metaheurística por entornos ansiosa se detiene el recorrido cuando se encuentre una solución del entorno mejor que la actual, pero el recorrido se continua de forma exhaustiva si no se encuentra tal mejora.

El solución desde el que comenzar el recorrido del entorno en la estrategia ansiosa es de gran importancia para aumentar la capacidad de exploración del procedimiento. Frente a la elección al azar de esta solución, una mejora del poder de explotación de la búsqueda se obtiene si las primeras soluciones vecinas examinadas son las más prometedoras. Las metaheurísticas *golosas* procuran que las primeras soluciones vecinas evaluadas tengan la mayor probabilidad posible de producir una mejora o que ésta sea de la mayor magnitud posible. El procedimiento se puede implementar usando alguna ordenación del entorno de la solución atendiendo a un análisis de la posible mejora producida por los movimientos mediante una estimación de la calidad de las nuevas soluciones.

3.3. Búsquedas globales

El principal inconveniente de las búsquedas locales es que si se alcanza una solución localmente óptima u óptimo local (una solución que es mejor que cualquiera de las de su entorno) la solución actual queda atrapada en su entorno [YI02], [AL96]. La regla de parada en las búsquedas monótonas implica detectar los mínimos locales analizando cuando no se mejora la solución actual. Una búsqueda con una perspectiva global del espacio de soluciones debe buscar herramientas para escapar de estas situaciones. Las principales metaheurísticas de búsqueda global surgen de las tres formas principales de escapar de esta situación: a) volver a comenzar la búsqueda desde otra solución inicial, b) modificar la estructura de entornos, y c) permitir movimientos que empeoran la solución actual.

Estas tres opciones dan lugar, respectivamente, a la metaheurística con arranque múltiple, a la metaheurística de entorno variable y a las metaheurísticas de búsqueda no monótonas. La tercera de las opciones incluye diversas metaheurísticas relevantes entre las que destacan la búsqueda probabilística, representada fundamentalmente por el Recocido Simulado (*Simulated Annealing*), y la búsqueda con memoria o Búsqueda Tabú (*Tabu Search*).

3.3.1. Búsqueda multi-arranque

Los procedimientos de búsqueda con *Arranque Múltiple* (*Multi-Start*) realizan varias búsquedas monótonas partiendo de diferentes soluciones iniciales [BK87], [Glo00], [Mar03], [MM03]. La búsqueda monótona implicada puede ser cualquiera de las anteriormente descritas. Una de las formas más simples de llevar esto a cabo consiste en generar una muestra de soluciones iniciales o de arranque. Esto es equivalente a generar al azar una nueva solución de partida cada vez que la búsqueda quede estancada en el entorno de una solución óptima local.

3.3.2. Búsqueda por entornos variables

La Búsqueda por *Entornos Variables* (*Variable Neighborhood Search*, VNS) es una metaheurística reciente que consiste en cambiar de forma sistemática la estructura de entorno [HM01], [HM], [HM02], [HM03], [HMM03]. La idea original fue considerar distintas estructuras de entornos y cambiarlas sistemáticamente para escapar de los mínimos locales. El VNS básico obtiene una solución del entorno de la solución actual, ejecuta una búsqueda monótona local desde ella hasta alcanzar un óptimo local, que reemplaza a la solución actual si ha habido una mejora y modifica la estructura de entorno en caso contrario. Una variante de esta estrategia básica es la búsqueda descendente por entornos variables (VND) aplica una búsqueda monótona por entornos cambiando de forma sistemática la estructura de entornos cada vez que se alcanza un mínimo local. Otra es utilizar una selección de los entornos, por medio de la elección de un conjunto de sus atributos (variables), de tal manera que se produzca una descomposición, y por tanto se definen determinados subproblemas, este modelo es denominado búsqueda por descomposición de entornos variables (VNDS).

3.3.3. Búsquedas no monótonas

Además de reiniciar la búsqueda y modificar la estructura de entornos, la otra vía para evitar quedarse atrapados en un óptimo local es admitir la posibilidad de pasos de no mejora, lo que da lugar a las estrategias de búsqueda no monótonas. Las metaheurísticas proponen principalmente controlar la aceptación de movimientos que no sean de mejora para que, al menos a la larga, se vayan mejorando las soluciones encontradas, y utilizar información histórica del proceso de búsqueda para controlar cuando el recorrido se está estancando en un mínimo local y evitar la formación de ciclos. Las metaheurísticas fundamentales que aplican estas estrategias de refinamiento son el recocido simulado, estrategia de búsqueda probabilística y la búsqueda tabú, estrategia de búsqueda con memoria.

Con las *metaheurísticas de búsqueda probabilísticas* se selecciona aleatoriamente un vecino de la solución actual que la reemplaza con cierta probabilidad. Por ejemplo, con probabilidad 1 si tiene mejor valor objetivo, y con una probabilidad menor que 1 si su valor objetivo es peor. Si el número de iteraciones es elevado, la búsqueda puede escapar de cualquier óptimo local si la probabilidad de aceptar peores soluciones va decreciendo. Generalmente la probabilidad de aceptar una solución peor es función del empeoramiento, de forma que, a menor diferencia en el valor objetivo, hay mayor probabilidad de ser aceptada.

El *Recocido Simulado* [KGV83], [LA87], [Vid93], [DD03] es el caso más importante de las metaheurísticas de búsqueda global con criterio de aceptación probabilístico. Es una secuencia de procedimientos de búsqueda que usa una probabilidad de aceptación de nuevas soluciones peores, que es función exponencial de la modificación de la función objetivo. Otras metaheurísticas simplemente reducen o incrementan esta probabilidad para modular la exploración y explotación de la búsqueda. Una variante del recocido simulado, la cual permite la optimización con respecto a una función de coste escalar es la *Metaheurística de Umbrales de Aceptación* (*Threshold Accepting*) [DS90], [Due93] que aceptan las nuevas soluciones peores que no sobrepasen el umbral y modulan este umbral con el mismo propósito. Ha sido ampliamente usada como procedimiento

de resolución de problemas estadísticos y econométricos.

Las *metaheurísticas de búsqueda con memoria* cuyo mejor representante es la *Búsqueda Tabú* comprenden las estrategias que tratan de utilizar la memoria del proceso de búsqueda para mejorar su rendimiento. Está fundamentada en las ideas expuestas por F. Glover en 1986 [Glo86] que ha contribuido con diversos trabajos [Glo89], [Glo90], [GW93], [GTW93], [GLTW93], [GL97] así como lo han hecho otros muchos autores en una extensa relación de artículos. En el origen del método el propósito era sólo evitar la reiteración en una misma zona de búsqueda recordando las últimas soluciones recorridas. Sin embargo, posteriormente se han realizado diversas propuestas para rentabilizar la memoria a medio o largo plazo.

La forma más directa de introducir la memoria en el procedimiento de búsqueda no monótono es considerar una función de aceptación que tenga en cuenta la historia de la búsqueda. El procedimiento elemental de Búsqueda tabú evita la repetición prematura de las mismas soluciones en el recorrido, para lo que prohíbe que las últimas soluciones vuelvan a utilizarse en el recorrido de búsqueda. Se utiliza un parámetro t que determina el número de últimas soluciones que son temporalmente prohibidas como nuevas soluciones actuales.

Estas estrategias se pueden aplicar dentro de la estructura de la búsqueda general de dos formas: introduciendo una función de aceptación que determine cuándo se acepta la nueva solución generada o modificando el procedimiento de generación del movimiento a aplicar a la solución actual. Con la primera de estas alternativas la función de aceptación puede incluir en sus parámetros información referente a la historia y el estado de la búsqueda, y a la solución generada. En el segundo caso, el procedimiento de generación de movimiento debe tener un diseño en el que se generan las soluciones vecinas de acuerdo con algún criterio que tenga en cuenta información de la historia y el estado de la búsqueda.

3.4. Búsquedas basadas en poblaciones o evolutivas

En una búsqueda en grupo o basada en poblaciones se sustituye la solución actual que recorre el espacio de soluciones, por un conjunto de soluciones que lo recorren conjuntamente interactuando entre ellas. Además de los movimientos aplicables a las soluciones que forman parte de este conjunto, denominado grupo o población de búsqueda, se contemplan otros operadores para generar nuevas soluciones a partir de las ya existentes.

3.4.1. Algoritmos genéticos

Las estrategias de búsqueda en grupo se iniciaron con el famoso *Algoritmo Genético* propuesto en [Hol75]. En la actualidad adoptan diversas características cómo se puede observar en la gran cantidad de trabajos editados sobre este tipo de procedimientos [BP92], [Dav91], [Gol89], [Mic92], [Mit96] y [Ree03]. A continuación se describen los conceptos fundamentales de su implementación para la solución de problemas de optimización.

En primer lugar, se establece una codificación apropiada de las soluciones del espacio de búsqueda y una forma de evaluar la función objetivo para cada una de estas codificaciones. Las soluciones se identifican con individuos que pueden formar parte de la población de búsqueda. La codificación de una solución se

interpreta como el cromosoma del individuo compuesto de un cierto número de genes a los que les corresponden ciertos alelos. Se consideran dos operaciones básicas: la mutación y el cruce. La mutación de un individuo consiste en modificar un gen cambiando, al azar, el alelo correspondiente. El cruce de dos individuos (llamados padres) produce un individuo hijo tomando un número k (elegido al azar) de genes de uno de los padres y los $t - k$ del otro. La población evoluciona de acuerdo a estrategias de selección de individuos, tanto para las operaciones como para la supervivencia. La selección se puede hacer simulando una lucha entre los individuos de la población con un procedimiento que, dados dos individuos se selecciona uno de ellos teniendo en cuenta su valoración (la función objetivo) y la adaptación al ambiente y a la población (criterios de diversidad, representatividad). La lucha por la supervivencia tiene por objeto mantener controlado el tamaño de la población. La selección de los luchadores se puede hacer de diferentes maneras: dos individuos seleccionados al azar, cada nuevo individuo con otro seleccionado al azar o con el peor de los existentes, etc.

Entre las metaheurísticas derivadas de los algoritmos genéticos destacan los *Algoritmos meméticos* [PM03], [MCP03], [Mos99] que surgen de combinar los algoritmos genéticos con búsquedas locales.

3.4.2. Algoritmos de estimación de distribuciones

Los *Algoritmos de Estimación de Distribuciones (EDA)* [LL02], [LLM03] son algoritmos evolutivos que usan una colección de soluciones candidatas para realizar trayectorias de búsqueda evitando mínimos locales. Estos algoritmos usan la estimación y simulación de la distribución de probabilidad conjunta como un mecanismo de evolución, en lugar de manipular directamente a los individuos que representan soluciones del problema. Un algoritmo *EDA* comienza generando aleatoriamente una población de individuos. Se realizan iterativamente tres tipos de operaciones sobre la población. El primer tipo de operación consiste en la generación de un subconjunto de los mejores individuos de la población. En segundo lugar se realiza un proceso de aprendizaje de un modelo de distribución de probabilidad a partir de los individuos seleccionados. En tercer lugar se generan nuevos individuos simulando el modelo de distribución obtenido. El algoritmo se detiene cuando se alcanza un cierto número de generaciones o cuando el rendimiento de la población deja de mejorar significativamente.

3.4.3. Búsqueda dispersa

El enfoque de la metaheurística de *Búsqueda Dispersa* (o *Scatter Search*) [LM02b], [LGM03], [ML03] contempla el uso de un conjunto de referencia de buenas soluciones dispersas. Este conjunto sirve, tanto para conducir la búsqueda, mejorando las herramientas para combinarlas adecuadamente, como para mantener un grado satisfactorio de diversidad. La propuesta inicial se originó en estrategias para crear reglas de decisión compuestas [Glo98]. Algunos estudios recientes demuestran las ventajas prácticas de este enfoque para resolver diversos problemas de optimización clásicos y reales. La Búsqueda dispersa se distingue de otros procedimientos en los mecanismos de intensificación y diversificación que explotan la memoria adaptada recurriendo a los fundamentos que unen la búsqueda dispersa a la búsqueda tabú.

3.4.4. Búsqueda por reencadenamiento de camino

El *reencadenamiento de camino* (PR, *Path Relinking*) [Glo98], [LGM00], [LGM03] es una metaheurística asociada a la búsqueda dispersa que utiliza la información que se obtiene de las mejores soluciones. Esta información se aprovecha en la mejora de otras soluciones que se encuentran posteriormente. Básicamente se trata de generar soluciones explorando las trayectorias que conectan soluciones de alta calidad. Partiendo de una de estas soluciones se genera un camino de soluciones hacia la otra solución incorporando a la primera atributos de la segunda. Este camino se construye tomando cada vez el atributo de la segunda solución que lo hace más cercano a ella. A continuación se toman, como puntos de arranque para nuevas fases de mejora, una o varias de las soluciones del recorrido anterior.

3.4.5. Algoritmos bionómicos

Otra estrategia de interés muy similar a las anteriores son los denominados *Algoritmos Bionómicos* (*Bionomic Algorithms*) que fueron introducidos por Christifides (1994) [Chr94] y ha sido aplicado a problemas logísticos [MMB98] y financieros [CCC03]. Se trata de una estrategia de optimización global evolutiva de búsqueda probabilística, similar a los algoritmos genéticos y a la búsqueda dispersa, que actualiza la población de soluciones en cada iteración, pero con una forma diferente en la búsqueda del espacio de soluciones. Inicialmente se genera una población inicial de soluciones. Esta población de soluciones iniciales puede ser generada de forma aleatoria. Después se mejora cada solución de la población inicial. Para ello se calcula el valor de cada solución considerada y un vector que mide la no factibilidad de la solución. Si la solución es no factible se aplica un algoritmo de optimización local buscando la factibilidad, si después de aplicar el algoritmo la solución obtenida sigue siendo no factible se pasa a otra solución. Si la solución si es factible lo que se busca es mejorar el valor de la solución con otro algoritmo de optimización local que mantenga la factibilidad. Se aplica una estrategia generacional, donde a partir de un conjunto de padres (generación previa) se generan hijos, con unos pasos específicos de madurez, mejora y crecimiento usando una heurística específica o procedimientos genéricos de búsqueda local y una aproximación estructurada para la selección de padres basada en la estructura de la población y la construcción de conjuntos de padres. Dos características importantes común con la búsqueda dispersa es la posibilidad de tener un conjunto de soluciones de tamaño variable y el uso de múltiples padres. En vez de usar una selección aleatoria de padres utiliza un procedimiento de identificación del conjunto independiente maximal de un grafo definido sobre el conjunto de soluciones.

Para generar nuevas soluciones se busca un conjunto de soluciones padres de forma que tengan un gran índice de dominancia y que entre ellas existan distinciones. La definición de distinción entre soluciones depende del problema. Después se usa un procedimiento que combina las soluciones padres para producir las soluciones hijos. Se calcula la dominancia de todos los hijos obtenidos, se elige el mejor y se mejora igual que en el caso de las soluciones de la población inicial mediante dos procedimientos de optimización local: uno para buscar la factibilidad si la solución no es factible y otro para mejorar el valor de la solución. Los empates en las preferencias se rompen de forma aleatoria. Se estudia si

este hijo mejora a algunas soluciones existentes y se reemplaza la solución más dominada por éste mismo hijo. El nuevo individuo también puede reemplazar a alguna solución vieja; la definición de solución vieja también es dependiente del problema. Si no se dan ninguna de estas dos circunstancias se rechaza al hijo. Además debe incorporar algún criterio de finalización.

3.5. Otras metaheurísticas de búsqueda

Se han propuesto otras metaheurísticas de búsqueda de cierta relevancia, algunas de las cuales presentan como novedad estar inspiradas en distintos fenómenos de la naturaleza, es decir bioinspiradas. Entre ellas destacan las redes neuronales, las colonias de hormigas, las bandadas de aves o bancos de peces. Otras metaheurísticas de búsqueda tienen el mérito de aplicar conceptos muy exitosos en otros campos de la IA, como el de las técnicas difusas en la metaheurística FANS o los métodos inteligentes de realizar búsqueda locales. Algunas de ellas están fundamentadas en paradigmas totalmente novedosos (por lo que se incluyen dentro del apartado de otras metaheurísticas), pero en la práctica la mayoría son estrategias que además de seguir utilizar patrones de búsqueda, utilizan conceptos de otras metaheurísticas y por tanto podemos decir que son metaheurísticas de búsqueda con componentes híbridas.

3.5.1. Optimización de partículas inteligentes

La *Optimización de Partículas Inteligentes* (PSO, *Particle Swarm Optimization*) [KE01] es una metaheurística evolutiva inspirada en el comportamiento social de las bandadas de pájaros o bancos de peces. Las soluciones, llamadas partículas se “echan a volar” en el espacio de búsqueda guiadas por la partícula que mejor solución ha encontrado hasta el momento y que hace de líder de la bandada. Cada partícula evoluciona teniendo en cuenta la mejor solución encontrada en su recorrido y al líder. El procedimiento también tiene en cuenta el mejor valor alcanzado por alguna de las partículas en su entorno. En cada iteración, las partículas modifican su velocidad hacia la mejor solución de su entorno teniendo en cuenta la información del líder. En ²³ hay más de 180 referencias sobre PSO.

3.5.2. Búsqueda por ruido

La *Búsqueda por Ruido* (*Nosing Method*) es un procedimiento diseñado por Charon y Hudry [CH93]. Estas metaheurísticas utilizan un tipo de perturbación o “ruido” para alterar aleatoria y sistemáticamente elementos del problema [CH01], [Sal97]. Está basado en el concepto de perturbación de datos (ruido), modificaciones en los valores de la solución. El procedimiento es una búsqueda local monótona, que partiendo de un solución inicial, se ejecuta algún método de perturbación de datos por el cual se van obteniendo cambios en los valores que toma la función objetivo y se itera el proceso. La intensidad de la perturbación se vá reduciendo continuamente hasta llegar a cero.

²³<http://web.ics.purdue.edu/~7Ehux/bibliography.shtml>

3.5.3. Búsqueda local iterada

La *Búsqueda Local Iterada* (*ILS, Iterated Local Search*) [Stü99], [LMS03] es una metaheurística que propone una estrategia exploratoria de mejora de las soluciones obtenidas por una determinada heurística mediante la iteración. Un esquema en el se incluye una heurística base que mejora los resultados mediante la repetición de dicha heurística. Esta idea ha sido propuesta en la literatura con distintas denominaciones, como descenso iterado, grandes pasos con cadenas de Markov, Lin-Kerningán iterado, búsqueda perturbada o ruidosa o la búsqueda de entorno variable con agitación donde la solución aportada por una heurística de búsqueda por entornos es *agitada* para producir una solución de partida para la heurística de búsqueda. La estrategia ILS actúa de la siguiente forma: dada una solución obtenida por la aplicación de la heurística base, se aplica un cambio o alteración que da lugar a una solución intermedia. La aplicación de la heurística base a esta nueva solución aporta una nueva solución que, si supera un test de aceptación, pasa a ser la nueva solución alterada. Aunque la heurística base incluida suele ser una búsqueda local, se ha propuesto aplicar cualquier otra estrategia, determinística o no. De esta forma, el proceso se convierte en una búsqueda estocástica por entornos donde los entornos no se explicitan sino que vienen determinados por la heurística base. Esta metaheurística se suele utilizar como entorno o incluida como componente de otras.

3.5.4. Heurística de concentración

La metaheurística de *Concentración* (*Concentration Heuristics*) [RH02] trata de combinar la información proporcionada por soluciones de calidad para realizar búsquedas locales. Básicamente consiste en, una vez obtenido un conjunto de concentración formado por buenas soluciones, abordar la búsqueda en una zona restringida a partir de la información proporcionada por dicho conjunto en el que se concentra la heurística.

3.5.5. Búsqueda local guiada

La metaheurística de *Búsqueda Local Guiada* (*GLS, Guided Local Search*) es una estrategia que guía la exploración del espacio de búsqueda con información de la estructura del problema y del propio proceso de búsqueda. Consiste básicamente en una la iteración de un procedimiento de búsqueda local y en la modificación de la función objetivo cada vez que termina la búsqueda local, mediante la inclusión de términos de penalización. Es decir penalizando determinados elementos que aparecen en el óptimo local obtenido, que son nuevas restricciones que definen mejor el problema, información que antes era desconocida para implementarla en la función objetivo [VT99], [MTF03], [VT03]. Estas produce una reducción de las soluciones a considerar y estimula la diversificación de la búsqueda. En este sentido se pueden considerar estrategias similares a la búsqueda tabú [Vou97].

3.5.6. Metaheurística FANS

La *Metaheurística de Búsqueda Difusa Adaptativa por Entornos* (*FANS, Fuzzy Adaptive Neighborhood Search*) [BPV02], [BPV03] es un procedimiento de búsqueda por entornos variables, que utiliza valoraciones borrosas o difusas

para medir el grado con que se consideran las soluciones con ciertas propiedades y esto es lo que usa para modificar la estructura de entorno. Se trata de una estrategia de búsqueda iterativa que incluye un mecanismo para construir nuevas soluciones, una función de evaluación difusa para evaluar las nuevas soluciones y un generador de entornos difusos, para generar las nuevas soluciones posibles.

3.5.7. Búsqueda reactiva

La *Búsqueda Reactiva* (*Reactive Search*) [Bat96], [BT94] es una metaheurística que propone usar, dentro de la búsqueda tabú, la información a largo plazo obtenida del recorrido. Se persigue detectar indicios de que la búsqueda necesita incrementar su exploración, por la repetición de ciertas estructuras o patrones en las soluciones recientemente visitadas. Esta información se almacena y se accede a ella utilizando técnicas eficientes de dispersión (*hashing*) o de árboles de búsqueda usuales en gestión de grandes cantidades de datos. Según la información que se tenga almacenada, en cada iteración se activa un proceso reactivo para alejarse de la zona de estancamiento.

3.5.8. Algoritmos Meméticos

La idea básica de los Algoritmos Meméticos es combinar mejoras individuales de las soluciones junto con procesos de cooperación y competiciones de tipo poblacional [PM03], [MCP03], [Mos99]. El término *memético* viene del término inglés *meme* interpretado como el concepto análogo del gen en el contexto de la evolución cultural.

Un Algoritmos Meméticos mantiene en todo momento una población de diversas soluciones al problema considerado llamadas *agentes*. Los agentes se relacionan entre sí por medio de la competición y la cooperación. La población de agentes evoluciona en base a generaciones; cada generación consiste en la actualización de la población mediante la recombinación de las características de algunos agentes seleccionados. La selección se encarga de elegir una muestra de los mejores agentes de la población actual mediante una función guía que evalúa la calidad de los agentes. La actualización tiene carácter competitivo, limitando el tamaño de la población, al eliminar algunos agentes para permitir otros nuevos. La función guía también puede utilizarse para seleccionar los agentes que se eliminan.

La creación de nuevos agentes se realiza en la fase de reproducción mediante la aplicación de operadores reproductivos; típicamente recombinación y mutación. La recombinación construye nuevos agentes mediante la cooperación entre agentes recombinados. Para la mutación se usan, básicamente, optimizadores locales basados en la aplicación iterativa de un operador de mutación arbitrario.

4. Otras metaheurísticas

Se han propuesto otras metaheurísticas de cierta relevancia, algunas de las cuales presentan como novedad estar inspiradas en distintos fenómenos de la naturaleza, es decir bioinspiradas. Entre ellas destacan las redes neuronales o las colonias de hormigas. Otras metaheurísticas tienen el mérito de aplicar conceptos muy exitosos en otros campos de la IA, como el de conjuntos difusos en la metaheurística FANS o técnicas de aprendizaje automático. La mayoría de ellas

están fundamentadas en paradigmas totalmente novedosos, pero en la práctica algunas son estrategias que además de seguir utilizar patrones de búsqueda, utilizan conceptos de otras metaheurísticas y por tanto podemos decir que son metaheurísticas de búsqueda con componentes híbridas.

4.1. Redes neuronales

Las *Redes Neuronales Artificiales* [Loo92] surgieron como modelos abstractos de sistemas nerviosos naturales formados por unidades de cómputo, llamadas neuronas, interconectadas. Estos modelos tienen la capacidad de ajustar sus parámetros en respuesta a unas entradas y salidas mejorando alguna función. Asociando los estados de la red a soluciones de un problema y utilizando el objetivo como referente, consiguen aproximarse al estado que corresponde con la solución óptima. La mayoría de las redes neuronales aplicadas para resolver problemas de optimización son versiones de la red de Hopfield [HT85]. La red de Hopfield puede autoajustarse para alcanzar el estado de mínima energía. La idea básica consiste en transformar el problema de optimización en la minimización de la función de energía de la red de Hopfield y determinar la estructura de una red neuronal de forma que las situaciones de energía mínima correspondan al estado de equilibrio de la red. De esta forma, la red evoluciona hacia el estado de equilibrio proporcionando la solución del problema. La principal ventaja de las redes se obtiene cuando, tras resolver el problema y disponer del estado de la red correspondiente, una modificación del modelo se traduce en una modificación de la red que provoca un rápido reajuste del equilibrio proporcionando la nueva solución al problema. Otras ventajas de las redes neuronales al resolver problemas combinatorios son su paralelización y la posibilidad de usar hardware específico. Otros modelos basados en redes neuronales aplicadas con éxito a problemas de optimización combinatoria son las máquinas de Boltzman y las redes competitivas WTA. Las máquinas de Boltzmann son un híbrido entre una red de Hopfield y la técnica de recocido simulado [AK89]. Las redes del tipo WTA (*Winner-Take-All*) [XJL⁺02] son modelos de redes neuronales competitivas que seleccionan de un conjunto de candidatos el elemento que maximiza el valor de activación siguiendo un sistema competitivo. Una revisión de la literatura de la aplicación de redes neuronales a problemas de optimización puede encontrarse en [Loo92] y [DGG⁺96].

4.2. Colonias de hormigas

La metaheurística de *Sistemas de Hormigas (Ant Systems)* empleada estrategias inspiradas en el comportamiento colectivo de las colonias de hormigas para descubrir fuentes de alimentación, estableciendo el camino más corto del hormigero hasta la fuente de alimentación. Cuando las hormigas se mueven inicialmente lo hacen al azar, dejando por el camino una cierta cantidad de feromonas, las siguientes las hacen por el camino que mayor cantidad de feromonas tenga. Se trata de una estrategia de búsqueda distribuida en la que un conjunto de agentes coopera en la búsqueda, transmitiendo e intercambiando información al resto. La idea inicial parte de los trabajos de M. Dorigo [DMC96], [BDT99], [DS03], [DS], aunque posteriormente otras variantes del inicial aparecen como híbridos usando algoritmos de aprendizaje y estrategias de búsqueda local son las más conocidas [DG96], [SH96],[SH97]. En estas estrategias se es-

tablece además de los procedimientos de búsqueda, iniciados con una población de soluciones iniciales que van recorriendo caminos a través de movimientos iterativos en su entorno de soluciones, mecanismos para almacenar y actualizar la información de los caminos de búsqueda recorridos que sirven para orientar las búsquedas hacia los mejores recorridos, así como funciones que especifican la intensidad de las feromonas en el tiempo y la persistencia de la información almacenada (feromonas) y algún criterio de parada.

4.3. Optimización extrema

La optimización *extrema* o extremal (EO, *Extreme Optimization*) [BP00] es una metaheurística inspirada en procesos auto-organizativos frecuentemente encontrados en la naturaleza. La idea central es utilizar modelos de evolución de ecosistemas que, en lugar de seleccionar los mejores elementos, llevan a la extinción a las componentes mal adaptadas del sistema. La idea básica del método es eliminar sucesivamente las componentes extremadamente indeseables de las soluciones subóptimas. En el modelo de evolución del ecosistema, cada especie i está caracterizada por el valor de su adaptación $\lambda_i \in \{0, 1\}$. La especie más débil, y las especies más estrechamente dependientes a ella, son sucesivamente seleccionadas para cambios adaptativos actualizando el valor de su adaptación. La optimización extrema opera con unos niveles de adaptación definidos para cada atributo de la solución que está directamente relacionada con la función objetivo. El método actúa sobre una única solución, y no sobre un conjunto de soluciones o población como los algoritmos genéticos, modificando el atributo de menor nivel de adaptación (y aquellos afectados por este cambio) aplicando algún tipo de transformación o movimiento.

4.4. Metaheurísticas GRASP

4.5. Satisfacción de restricciones

La *Satisfacción de Restricciones* (*Constraint Satisfaction*) [Dec03], [MG03], [FW03] es un paradigma usado para resolver diversos tipos de problemas, expresados como, encontrar las soluciones, los valores de un conjunto de variables en dominios específicos que satisfacen un conjunto de restricciones sobre estas variables. Muchos problemas de optimización combinatoria se pueden expresar de esta manera [PG96], [Hei99]. Este enfoque puede considerarse un paradigma general propio dentro de las metaheurísticas, donde lo más relevante es la atención que se le presta al procesamiento de las restricciones que surgen en un problema y como afecta a los procedimientos de búsqueda de soluciones. En los últimos años ha tenido un desarrollo importante por la facilidad para modelizar problemas complejos, así como por la flexibilidad en el tratamiento de las restricciones. Fruto de ello, es la generalización del paradigma en modelización de problemas y en el tratamiento declarativo de su implementación, para la resolución de los mismos, denominado la *Programación con Restricciones* (*Constraint Programming*) [MS98] y los avances en entornos de desarrollo estandarizados como OPL, SALSA y otros. Las estrategias generales de solución en este paradigma pertenecen dos categorías las que razonan sobre las propias restricciones procedimientos de propagación de restricciones o técnicas de consistencia y la otra basada en la exploración del espacio de búsqueda con métodos de búsqueda.

da constructiva, a partir de la asignación sistemática de valores parciales a las variables hasta completar la solución, con algoritmos conocidos de vuelta atrás y también algoritmos de ramificación y acotación para los problemas de optimización, o también mediante exploraciones del espacio de soluciones mediante heurísticas de búsqueda local y otras metaheurísticas [Dec98], [Bar99], [Fro97]. Las metaheurísticas tratadas han encontrado un amplio espacio de aplicación en este contexto, donde la combinación adecuada de las técnicas de consistencia con metaheurísticas han producido resultados óptimos en diversos problemas [CEM00].

4.6. Metaheurísticas multiobjetivo

Un aspecto importante de la visión global de las metaheurísticas es su aplicación en la resolución de problemas *multiobjetivo* [Teg01], [Han00], el creciente interés de la optimización que implica tomar decisiones sobre varios objetivos a la vez, es decir con múltiples funciones objetivos, ha motivado la aplicación de las metaheurísticas tratadas anteriormente, usadas para resolver problemas con una función objetivo, sobre problemas de optimización de este tipo. Esto supone la adaptación de dichas metaheurísticas. La mayoría de las metaheurísticas multiobjetivo están basadas en algoritmos genéticos, recocido simulado y en búsquedas tabú [Han98], [FF95], aunque hay otras como GRASP [GVT98], las razones de uso de las primeras es el que trabaja con poblaciones de soluciones y se puede convertir fácilmente la búsqueda para múltiples soluciones en paralelo, o se puede repetir la ejecución de las metaheurísticas para una función objetivo, con una combinación de funciones, estableciendo pesos para cada una de ellas que pueden cambiar en cada ejecución.

4.7. Metaheurísticas paralelas

Otro aspecto de interés en la visión global de las metaheurísticas es la aplicación de estrategias de *paralelización* de metaheurísticas particulares y sobre cuestiones genéricas de metaheurísticas paralelas [Aze92], [CMRR01]. Estas técnicas se han utilizado con la finalidad de conseguir mejoras en el rendimiento de la metaheurística en la resolución de problemas de optimización especialmente duros utilizando recursos computacionales óptimos.

4.8. Hiperheurísticas

5. Propiedades deseables

En esta sección analizamos un conjunto de propiedades *deseables* de las metaheurísticas. Son propiedades deseables todas aquellas que favorezcan el interés práctico y teórico de las metaheurísticas. Indicarán direcciones a las que dirigir los esfuerzos para contribuir al desarrollo científico e ingenieril, pero no será posible mejorar todas las propiedades a la vez, dado que algunas son parcialmente contrapuestas. Una relación de tales propiedades debe incluir las siguientes:

- **Simple.** La metaheurística debe estar basada en un principio sencillo y claro; fácil de comprender.

- *Precisa.* Los pasos y fases de la metaheurística deben estar formulados en términos concretos.
- *Coherente.* Los elementos de la metaheurística debe deducirse naturalmente de sus principios.
- *Efectiva.* Los algoritmos derivados de la metaheurística deben proporcionar soluciones de muy alta calidad; óptimas o muy cercanas a las óptimas.
- *Eficaz.* La probabilidad de alcanzar soluciones óptimas de casos realistas con la metaheurística debe ser alta.
- *Eficiente.* La metaheurística debe realizar un buen aprovechamiento de recursos computacionales; tiempo de ejecución y espacio de memoria.
- *General.* La metaheurística debe ser utilizable con buen rendimiento en una amplia variedad de problemas.
- *Adaptable.* La metaheurística debe ser capaz de adaptarse a diferentes contextos de aplicación o modificaciones importantes del modelo.
- *Robusta.* El comportamiento de la metaheurística debe ser poco sensible a pequeñas alteraciones del modelo o contexto de aplicación.
- *Interactiva.* La metaheurística debe permitir que el usuario pueda aplicar sus conocimientos para mejorar el rendimiento del procedimiento.
- *Múltiple.* La metaheurística debe suministrar diferentes soluciones alternativas de alta calidad entre las que el usuario pueda elegir.
- *Autónoma.* La metaheurística debe permitir un funcionamiento autónomo, libre de parámetros o que se puedan establecer automáticamente.
- *Flexible.* La metaheurística debe permitir el ajuste flexible de sus parámetros y criterios, así como el manejo flexible de las restricciones del problema.
- *Equilibrada.* La metaheurística debe mantener un equilibrio permanente entre la búsqueda de soluciones de calidad y el uso de recursos computacionales.
- *Modelable.* La metaheurística debe permitir con el menor esfuerzo su modelización, diseño e implementación a la hora de resolver problemas reales y concretos como para facilitar su aplicación e integración en entornos de desarrollo más generales.

Varias de estas propiedades están muy relacionadas y apuntan en la misma dirección, como la simplicidad, la precisión y la coherencia. La *simplicidad* de la metaheurística facilita su uso y contribuye a dotarla de amplia aplicabilidad. La descripción formal de las operaciones debe liberarse de la analogía física o biológica que haya sido la fuente inicial de inspiración para permitir mejoras que no respeten la analogía. La *precisión* en la descripción de los elementos que componen la metaheurística es crucial para concretar un procedimiento de alta calidad; fácil de implementar. Los pasos de los procedimientos básicos de los algoritmos deben traducirse coherentemente de los principios en que se inspira. Debe huirse de sentencias sin sentido o vagas. Frecuentemente se presentan como extensiones de una metaheurística la incorporación de herramientas o recursos computacionales estándares, o de pautas de otras metaheurísticas cuando en realidad deben calificarse como hibridaciones de las mismas.

La evaluación del rendimiento de una metaheurística debe atender tanto a la eficiencia como a la efectividad y eficacia de los procedimientos heurísticos obtenidos. Para validar la *efectividad* y *eficacia* de una metaheurística, éstas deben afrontar con éxito problemas de un banco de casos reales para los que se conozcan las soluciones. Si no se dispone de estos casos, se deben construir recurriendo a procesos de simulación que se aproximen a tales circunstancias. La *eficiencia* del método se contrasta experimentalmente en el empleo de un tiempo computacional moderado (o al menos razonable) para alcanzar éxito en los problemas considerados. El tamaño de los problemas considerados en las aplicaciones prácticas de los métodos de optimización se limita por las herramientas disponibles para resolverlos más que por la necesidad de los potenciales usuarios. Cuando las metaheurísticas se aplican a instancias realmente grandes, sus fortalezas y debilidades aparecen más claramente. Las metaheurísticas pueden mejorar su rendimiento extendiéndose en varias direcciones y, posiblemente, hibridándose. Los procedimientos heurísticos resultantes se complican y usan muchos parámetros. Con ello se puede mejorar su eficiencia, pero enmascaran las razones de su éxito. En algunas ocasiones la alta especialización de una metaheurística lleva a un ajuste fino de parámetros sobre algún conjunto de entrenamiento concreto.

La aplicabilidad de una metaheurística debe estar sustentada en la generalidad, pero también en su adaptabilidad y robustez. La *robustez* tiene que ser contrastada experimentalmente analizando el rendimiento frente a fluctuaciones de las características de los problemas. La robustez se refleja en que el número de parámetros que hay que fijar en las distintas aplicaciones se mantiene bajo. La *generalidad* de una metaheurística se refleja en la diversidad de los campos de aplicación para los que se han utilizado con éxito. La *adaptabilidad* permite que las conclusiones obtenidas al afrontar un tipo de problemas particular puedan ser aprovechadas en otros contextos. Las pautas proporcionadas por una metaheurística de búsqueda se aplican a descripciones asociadas a un problema, referidas simplemente a los movimientos posibles para transformar una solución en otra y la forma de evaluarlas.

Para favorecer la utilidad de la metaheurística en la resolución de problemas reales, por ejemplo incorporándolo a Sistemas de Ayuda a la Decisión, son importantes las propiedades que propicien un *interface* amigable. La *interactividad* de los sistemas basados en las metaheurísticas favorece la colaboración con otros campos que proporcionan conocimientos específicos de los problemas para mejorar el rendimiento de la metaheurística. La posibilidad de ofrecer di-

versas soluciones de alta calidad, realmente diferentes, entre las que los decisores puedan optar contribuye a diseminar su uso. La relativa *autonomía* de implementaciones de la metaheurística permite ganarse la confianza de usuarios poco expertos en optimización o en los campos de aplicación.

Una característica que contribuye a divulgar una metaheurística es la novedad a la que va asociada, en cuanto a la originalidad de los principios que la inspiran y a los campos de repercusión social a los que se aplica. Este aspecto se revela, por ejemplo, en la inspiración en fenómenos naturales de los algoritmos genéticos y otras metaheurísticas, en la aplicación a la demostración matemática de la metaheurística de entorno variable, y en la aplicación a la ingeniería genética de las técnicas FANS. Sin embargo, en los entornos científicos, tecnológicos, ingenieril o empresarial, el aspecto más relevante es el éxito asociado a la eficiencia y efectividad de los algoritmos derivados de cada metaheurística en la resolución de problemas de gran tamaño o surgidos en aplicaciones reales.

6. Conclusiones

Para la resolución práctica de una proporción cada vez mayor de problemas de interés, no resulta apropiado utilizar procedimientos diseñados a propósito para cada modelo y dependientes de su estructura particular. Ante la necesidad de utilizar algoritmos heurísticos de solución, las metaheurísticas proporcionan pautas y estrategias generales de diseño para obtener heurísticas con un alto rendimiento. Las metaheurísticas proporcionan métodos para escaparse de los óptimos locales de mala calidad por lo que, dado que el valor de tales óptimos locales frecuentemente difiere considerablemente del valor del óptimo global, el impacto práctico de las metaheurísticas ha sido inmenso.

Se observan diversas tendencias en las investigaciones sobre técnicas metaheurísticas. Unas tratan de mantener la *pureza* de los métodos y comprobar su efectividad en nuevos problemas, sin incorporar herramientas de otras metaheurísticas. Otras investigaciones, desde una perspectiva más ingenieril, tratan de aprovechar los recursos proporcionados por cada una de ellas. Para estos últimos, la única cuestión relevante es conocer si el beneficio en el rendimiento, proporcionado por la inclusión de tales herramientas, compensa al esfuerzo de su implementación y al incremento de la complejidad de los códigos resultantes.

El campo de investigación sobre las metaheurísticas ofrece más oportunidades para aplicar la intuición que la deducción. En contraste con el éxito práctico de muchas metaheurísticas, el estudio teórico está más retrasado. Frecuentemente se obtienen buenas nuevas heurísticas, con algo de inventiva y gran esfuerzo en el ajuste de numerosos parámetros, pero las razones de por qué funcionan tan bien permanecen desconocidas. La situación es incluso peor para los híbridos, donde las aportaciones de las metaheurísticas implicadas y el beneficio de la interacción raramente son objetos de un estudio experimental bien diseñado.

Algunas propuestas encaminadas a una mejor comprensión de estos aspectos son el estudio de la influencia de la topografía de los óptimos locales y de las trayectorias seguidas por los procesos de búsqueda heurística. El análisis de la evolución de las distancias al óptimo frecuentemente se centran exclusivamente en la desviación del objetivo alcanzado frente al mejor posible. Se puede obtener información más útil si se consideran distancias entre las propias soluciones y

no sólo su valor.

Los intentos por organizar este campo son numerosos, pero los conceptos principales son raramente definidos con precisión y hay todavía muy pocos teoremas significativos. Ninguna estructura ha conseguido una aceptación general. Más bien, cada grupo de investigación inspirador de una metaheurística tiene su propio punto de vista y habilidad para explicar muchas heurísticas en su propio vocabulario así como para absorber ideas de todo el campo (generalmente bajo la forma de híbridos).

La peor consecuencia de este hecho es la tendencia a la proliferación de reclamaciones de prioridades basadas en evidencias tan vagas que son difíciles de evaluar. Con algunos argumentos o la reutilización de términos en la descripción de unas metaheurísticas y otras, se puede interpretar que una de ellas es la otra definida de manera incompleta (si no se especifica algún elemento importante o es descrito por alguna vaga metáfora) o como un caso particular, al restringir el tipo de herramienta aplicada a un tipo de problema. Esto sería igualmente arbitrario. Parece que el carácter babélico de la investigación en metaheurísticas es, esperemos que temporalmente, ligeramente deshonesto. Mientras esto permanezca así, éxitos claros en problemas particulares serán más importantes para evaluar las metaheurísticas que largas controversias. Finalmente, cuando se consideren globalmente las cualidades deseables de las metaheurísticas, las comparativas de eficiencia no tendrían el papel tan dominante, algunas veces exclusivo, que se les da en muchos artículos. El propósito de estas investigaciones debe ser la comprensión de las metaheurísticas, no la competición entre ellas. Otras cualidades de las heurísticas y las metaheurísticas distintas que la eficiencia pueden ser tan importantes a la larga, como la simplicidad, la precisión, la robustez, y, sobre todo la, amigabilidad.

Cuadro 1: Relación de metaheurísticas.

SIGLAS	NOMBRE	TRADUCCIÓN
CP	Constraints Programming	Programación por Restricciones
EA	Evolutionary Algorithms	Algoritmos Evolutivos
EMOO	Evolutionary Multi-Objective Optimization	Optimización Evolutiva Multi-objetivo
GA	Genetic Algorithm	Algoritmos Genéticos
GRASP	Greedy Randomize Adaptative Search Procedure	Algoritmo GRASP
MA	Memetic Algorithms	Algoritmos Meméticos
MOMH	Multiobjective Optimization MetaHeuristics	MetaHeurísticas de Optimización Multiobjetivo
PMH	Parallel Metaheuristics	Metaheurísticas paralelas
SA	Simulated Annealing	Templado simulado
SS	Scatter Search	Búsqueda Dispersa
TS	Tabu Search	Búsqueda Tabú
VNS	Variable Neighbourhood Search	Búsquedas de Entorno Variable

El listado de metaheurísticas que contempla la red HEUR

La lista siguiente recoge los procedimientos más establecidos (hemos mantenido su acepción en inglés siguiendo el uso habitual).

- ACO Ant Colony Optimization
- ALSS Advanced Local Search Strategies
- CP Constraint Programming
- EA Evolutionary Algorithms
- EDA Estimation Distribution Algorithms
- FANS Fuzzy Adaptive Neighborhood Search
- GA Genetic Algorithms
- GLS Guided Local Search
- GP Genetic Programming
- GRASP Greedy Randomize Adaptive Search Procedures
- HC Heuristic Concentration
- ILS Iterated Local Search
- MA Memetic Algorithms
- MOS Multi-Objective Search
- MSM Multi-Start Methods
- NM Noising Methods
- NN Neural Networks
- PR Path Relinking
- SA Simulated Annealing
- SS Scatter Search
- TAM Threshold Accepting Methods
- TS Tabu Search
- VNS Variable Neighborhood Search

Referencias

- [Ack87] D. H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. Kluwer, 1987.
- [ACR98] A. Andreatta, S Carvalho, and C Riberiro. An object-oriented framework for local search heuristics. Technical report, Catholic University of Rio de Janeiro, 1998.
- [ACR99] A. Andreatta, S Carvalho, and C Riberiro. A framework for the development of local search heuristics for combinatorial optimization problems. 2nd Metaheuristics International conference, 1999.
- [AK89] E.H.L. Aarts and J. Korst. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. Wiley, 1989.
- [AL96] E.H.L. Aarts and J.K. Lenstra. *Local Search in Combinatorial Optimization*. Wiley, 1996.
- [Aze92] R. Azencott, editor. *Simulated annealing - parallelization techniques*. Wiley, 1992.
- [Bar99] R. Bartak. Constraint programming: In pursuit of the holy grail. In *In Proceeding of the WDS*, Progue, 1999.
- [Bat96] R. Battiti. Reactive search: towards self-tuning heuristics. In V.J. Rayward-Smith, I.H. Osman, C.R. Reeves, and G.D. Smith, editors, *Modern heuristic search methods*, pages 61–83. Wiley, 1996.
- [BDT99] E. Bonabeau, M. Dorigo, and T. Theraulaz. *From Natural to Artificial Swarm Intelligence*. Oxford University Press, 1999.
- [Bea93] J.E. Beasley. Lagrangian relaxation. In C.R. Reeves, editor, *Modern heuristic techniques for combinatorial problems*, pages 243–303. Blackwell Scientific Publications, 1993.
- [BFS99] B. De Backer, V. Furnon, and P. Shaw. An object model for meta-heuristic search in constraint programming. In *Proceeding of CP-AI-OR '99. Workshop on Integration of AI and OR techniques in CP for Combinatorial Optimization Problems*, 1999.
- [BK87] C.G.E. Boender and A.H.G. Rinnooy Kan. Bayesian stopping rules for multistart global optimization methods. *Mathematical Programming*, 37:59–80, 1987.
- [BMM03] J.A. Moreno-Pérez B. Melián and J.M. Moreno-Vega. Metaheurísticas: Una visión global. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:7–28, 2003.
- [BP92] B.P. Buckles and F.E. Petry. *Genetic Algorithms*. IEEE Computer Society Press, 1992.
- [BP00] S. Boettcher and A.G. Percus. Nature’s way of optimizing. *Artificial Intelligence*, 119:275–286, 2000.

- [BPV02] A. Blanco, D. Pelta, and J.L. Verdegay. A fuzzy valuation-based local search framework for combinatorial optimization. *Journal of Fuzzy Optimization and Decision Making*, 1(2):177–193, 2002.
- [BPV03] A. Blanco, D. Pelta, and J.L. Verdegay. Fans: una heurística basada en conjuntos difusos para problemas de optimización. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:103–122, 2003.
- [BR01] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. Technical report, IRIDIA, Bruxelles. Belgium, 2001.
- [BT94] R. Battiti and G. Tecchioli. The reactive tabu search. *ORSA Journal of Computing*, 6:126–140, 1994.
- [BY02] Hemant K. Bhargava and Nong Ye, editors. *Computational Modeling and Problem Solving in the Networked World*, volume 21 of *Interfaces in Computer Science and Operations Research*. Kluwer Academic Publishers, 2002.
- [CCC03] S. Christofides, A. Christofides, and N. Christofides. The design of corporate tax structure. 98:493–510, 2003.
- [CDG99] David Corne, Marco Dorigo, and Fred Glover, editors. *New Ideas in Optimization*. McGraw-Hill, 1999.
- [CEM00] B.G. Craenen, A.E. Eiben, and E. Marchiori. Solving constraint satisfaction problems with heuristics-bases evolutionary algorithms. In *In proceeding of the 2000 Congress on Evolutionary Computation*, pages 1569–1575, 2000.
- [CH93] I. Charon and O. Hudry. The noising method: A new method for combinatorial optimization. *Operations Research Letters*, 14:133–137, 1993.
- [CH01] I. Charon and O. Hudry. The noising methods: A generalization of some metaheuristics. *European Journal of Operational Research*, 135:86–101, 2001.
- [Chr94] N. Christofides. The bionomic algorithm. Savona. Italy, 94. AIRO.
- [CMRR01] V. Cung, S Martins, C Ribeiro, and C Roucairol. *Strategies for the Parallel Implementation of Metaheuristics*. Essays and Surveys in Metaheuristics, kluwer edition, 2001.
- [Con92] D. Connolly. General purpose simulated annealing. *Journal of Operational Research Society*, 43:495–505, 1992.
- [COS00] David W. Corne, Martin J. Oates, and George D. Smith, editors. *Telecommunications Optimization: Heuristic and Adaptive Techniques*. John Wiley & Sons, 2000.

- [CS01] J. Claro and J.P. Sousa. An object-oriented framework for multiobjective local search. Porto, Portugal, 2001. 4th metaheuristics International Conference, MIC'2001.
- [Dav91] L. Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
- [DBT99] M. Dorigo, E. Bonabeau, and T. Theraulaz. *From Natural to Artificial Swarm Intelligence*. Oxford University Press, 1999.
- [DD03] K. Dowsland and B.A. Díaz. Diseño de heurísticas and fundamentos del recocido simulado. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:93–102, 2003.
- [Dec98] R. Dechter. *Constraint Satisfaction*. In the MIT encyclopedia of the Cognitive Sciences, 1998.
- [Dec03] R. Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.
- [DG96] M. Dorigo and L. Gambardella. A study of some properties of ant-q. In Springer-Verlag, editor, *4th Int. Conf. on Parallel Problem from Nature*, pages 656–665, Berlin, 1996.
- [DGG⁺96] A. Daz, F. Glover, H. M. Ghaziri, J. L. González, M. Laguna, P. Moscato, and F. T. Tseng. *Optimización Heurística y Redes Neuronales*. Ed. Paraninfo, 1996.
- [DMC96] M. Dorigo, V. Maniezzo, and A. Colomi. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 26:1, 29–41, 1996.
- [DS] M. Dorigo and T. Stützle. *The Ant Colony Optimization*.
- [DS90] G. Dueck and T. Scheuer. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *Computational Physics*, 90:161–175, 1990.
- [DS03] M. Dorigo and T. Stützle. The ant colony optimization metaheuristic: Algorithms applications, and advances. In F. Glover and G. Kochenberger, editors, *Handbook on MetaHeuristics*. 2003.
- [Due93] G. Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104:86–92, 1993.
- [EBS] G. Kendall E.K. Burke and E. Soubeiga. A tabu-search hyperheuristic for timetabling and rostering.
- [FF95] C.M. Fonseca and P.J. Fleming. A overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3 (1):1–16, 1995.
- [FGK93] R. Fourer, D. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. The Scientific Press, San Francisco, CA, 1993.

- [FHL96] J. Ferland, A. Hertz, and A. Lavoie. An object-oriented methodology for solving assignment-type problems with neighborhood search techniques. *Operations Research*, 44 (2):347–359, 1996.
- [FLL01] F. Focacci, F. Laburthe, and A. Lodi. Local search and constraint programming. In *Proceedings of MIC'2001*, pages 451–454, Porto, Portugal, 2001. 4th Metaheuristics International Conference.
- [Fro97] D. Frost. *Algorithms and Heuristics for Constraint Satisfaction Problems*. PhD thesis, University of California, 1997.
- [FT61] H. Fisher and G. L. Thompson. Probabilistic learning combinations of local job-shop scheduling rules. In *Factory Scheduling Conference*, Carnegie Institute of Technology, 1961.
- [FV01] A. Fink and S. Voss. Reusable metaheuristic software components and their application via software generators. In *Proceedings of MIC'2001*, pages 637–641, Porto. Portugal, 2001. 4th Metaheuristics International Conference.
- [FVW98] A. Fink, S. Voss, and D. Woodruff. Building reusable software components for heuristic search. Zurich, 1998. OR98.
- [FVW99] A. Fink, S. Voss, and D. Woodruff. An adoption path for intelligent heuristic search componentware. In E. Rolland and n.S. Umanath, editors, *Proceedings of the 4th INFORMS Conference on Information Systems and Tecnology*, pages 153–168, Linthicum, 1999. INFORMIS.
- [FW03] E. Freuder and M. Wallace. Constraint satisfaction. In F. Glover and G. Kochenberger, editors, *Handbook on MetaHeuristics*, chapter 14. 2003.
- [GK03] F. Glover and G. Kochenberger, editors. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.
- [GL97] F. Glover and M. Laguna. *Tabu Search*. Kluwer, 1997.
- [Glo86] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5:533–549, 1986.
- [Glo89] F. Glover. Tabu search. part I. *ORSA Journal on Computing*, 1:190–206, 1989.
- [Glo90] F. Glover. Tabu search. part II. *ORSA Journal on Computing*, 2:4–32, 1990.
- [Glo98] F. Glover. A template for scatter search and path relinking. In J.-K. Hao and E. Lutton, editors, *Artificial Evolution*, volume 1363, pages 13–54. Springer-Verlag, 1998.
- [Glo00] F. Glover. Multi-start and strategic oscillation methods - principles to exploit adaptive memory. In M. Laguna and J.L. González-Velarde, editors, *Computing Tools for Modeling, Optimization and Simulation*, pages 1–24. Kluwer Academic Publishers, 2000.

- [GLTW93] F. Glover, M. Laguna, E.D. Taillard, and D. De Werra. Tabu search. *Annals of Operational Research*, 43, 1993.
- [GM03] F. Glover and B. Melián. Búsqueda tabú. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:29–48, 2003.
- [Gol89] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [GP99] M. Graccho and S. Porto. Taboobuilder: an object-oriented framework for building tabu search applications. Angra dos Reis, Brasil, 1999. MIC 99.
- [Gro92] L.K. Grover. Local search and the local structure of NP-complete problems. *Operational Research Letter*, 12:235–243, 1992.
- [GS01] L. D. Gaspero and A. Schaerf. Easylocal++: An object-oriented framework for design of local search algorithms and metaheuristics. In *4th Metaheuristics International Conference MIC'2001*, Porto, Portugal, 2001.
- [GS02] L. D. Gaspero and A. Schaerf. Easylocal++: An object-oriented framework for design of local search algorithms and metaheuristics. Porto, Portugal, 2002. 4th Metaheuristics International Conference.
- [GTW93] F. Glover, E. Taillard, and D. De Werra. A user's guide to tabu search. *Annals of Operations Research*, 41:3–28, 1993.
- [Gui02] M. Guignard. Lagrangian relaxation. In P.M. Pardalos and M.G.C. Resende, editors, *Handbook of Applied Optimization Oxford University Press*, pages 465–474. 2002.
- [GVT98] X. Gandibleux, D. Vancoppenolle, and D. Tuytens. A first making use of grasp for solving moco problems. Technical report, University Valenciennes, France, 1998.
- [GW93] F. Glover and D. De Werra. *Tabu Search*, volume 41. Baltzer, 1993.
- [Han98] M.P. Hansen. *Metaheuristics for Multiple Objectives Combinatorial Optimization*. PhD thesis, Institute Mathematical Modelling. Technical University of Denmark, 1998.
- [Han00] T. Hanne. Global multiobjective optimization using evolutionary algorithms. *Journal of Heuristics*, 6 (3):347–360, 2000.
- [Hei99] S. Heipcke. Comparing constraint programming and mathematical programming approaches to discrete optimisation- the change problem. *Journal of the Operational Research Society*, 50:581–595, 1999.
- [Hen99] P. Van Hentenryck. *The OPL Optimization Programming Language*. The MIT Press, Cambridge, 1999.
- [HK00] A. Hertz and D. Kobler. A framework for the description of evolutionary algorithms. *European Journal of Operational Research*, 126:1–12, 2000.

- [HM] P. Hansen and N. Mladenović. Developments in variable neighbourhood search. In C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 415–439.
- [HM01] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.
- [HM02] P. Hansen and N. Mladenović. Variable neighborhood search. In P.M. Pardalos and M.G.C. Resende, editors, *Handbook of Applied Optimization*, pages 221–234. Oxford University Press, 2002.
- [HM03] P. Hansen and N. Mladenović. Variable neighborhood search. In F. Glover and G.A. Kochenberger, editors, *Handbook of Metaheuristics*, chapter 6. Kluwer Academic, 2003.
- [HMM03] P. Hansen, N. Mladenović, and J.A. Moreno. Búsqueda de entorno variable. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:77–92, 2003.
- [Hol75] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [HT85] J.J. Hopfield and D.W. Tank. Neural computation of decisions in optimization problems. *Bio. Cybern.*, 52:141–152, 1985.
- [KE01] Y.S.J. Kennedy and R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, 2001.
- [KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [KS00] N. Krasnogor and J. Smith. Mafra: A java memetic algorithms framework (2000). In *2000 Genetic and Evolutionary Computation Conference, First International Workshop On Memetic Algorithms*, pages 125–130, Las Vegas, Nevada, USA, 2000.
- [LA87] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated Annealing: Theory and Applications*. Kluwer Academic Press, 1987.
- [Lag97] M. Laguna. Metaheuristic optimization with evolver, genocop and optquest. In J. Barcelo, editor, *EURO/INFORMS Joint International Meeting 1997 Plenaries and Tutorials*, pages 141–150, 1997.
- [LC98] F. Laburthe and Y. Caseau. Salsa: A language for search algorithms. In Springer, editor, *4th Int. Conf. on Principles and Practice of Constraint Programming CP98*, pages 310–324, Pisa Italy, 1998. Lecture Notes in computer Science 1152.
- [LG00] M. Laguna and J.L. González-Velarde. *Computing Tools for Modeling, Optimization and Simulation*. Kluwer Academic Publishers, 2000.
- [LGM00] M. Laguna, F. Glover, and R. Martí. *Control and Cybernetics*, 39:653–684, 2000.

- [LGM03] M. Laguna, F. Glover, and R. Martí. In F. Glover and G. Kochenberger, editors, *Handbook on MetaHeuristics*, chapter 1. Scatter search and path relinking: Advances and applications, 2003.
- [LL02] J.A. Lozano and P. Larrañaga. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic, 2002.
- [LLM03] P. Larrañaga, J.A. Lozano, and H. Mühlenbein. Algoritmos de estimación de distribuciones en problemas de optimización combinatoria. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:149–168, 2003.
- [LM02a] M. Laguna and R. Martí. The optquest callable library. In Stefan Voss and David L. Woodruff, editors, *Optimization Software Class Libraries*, pages 193–218. Kluwer Academic Publishers, 2002.
- [LM02b] M. Laguna and R. Martí. *Scatter Search Methodology and Implementations in C*, volume 24 of *OPERATIONS RESEARCH/COMPUTER SCIENCE INTERFACES*. Kluwer Academic Publishers, 2002.
- [LMS03] H.R. Lourenço, O. Martin, and T. Stützle. Iterated local search. In F. Glover and G.G. Kochenberger, editors, *Handbook of Metaheuristics*, chapter 11. Kluwer Academic Publishers, 2003.
- [LO96] G. Laporte and I.H. Osman, editors. *Metaheuristic in Combinatorial Optimization*, volume 63 of *Annals of Operational Research*. Baltzer Science Pub., 1996.
- [Loo92] C.K. Looi. Neural network method in combinatorial optimization. *Computers and Operations Research*, 19:191–208, 1992.
- [Mar03] R. Martí. Multistart methods. In Fred Glover and Gary A. Kochenberger, editors, *Handbook of Metaheuristics*, pages 355–368. Kluwer Academic, 2003.
- [MCP03] P. Moscato and C. Cotta-Porras. Una introducción a los algoritmos meméticos. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:131–148, 2003.
- [MF00] Z. Michalewicz and D.B. Fogel. *How to Solve It: Modern Heuristics*. Springer Verlag, 2000.
- [MG03] F. Manyá and C. Gomes. Técnicas de resolución de problemas de satisfacción de restricciones. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:169–188, 2003.
- [MH97] L. Michel and P. Van Hentenrick. Localizer: A modeling language for local search. In Springer, editor, *3rd int. Conf. on Principles and Practice of Constraint Programming CP97*, pages 238–252, Austria, 1997. Lecture Notes in Computer Science 1330.
- [MH99] L. Michel and P. Van Hentenrick. Localizer: A modeling language for local search. *INFORMS Journal of Computing*, 11:1–14, 1999.

- [MH01] L. Michel and P. Van Hentenryck. Localicer++: An open library for local search. Technical report, University de Brown, Providence, 2001.
- [Mic92] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 1992.
- [Mit96] M. Mitchel. *An introduction to Genetic Algorithms*. MIT Press, 1996.
- [ML03] R. Martí and M. Laguna. Scatter search: Diseño básico and estrategias avanzadas. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:123–130, 2003.
- [MM03] R. Martí and J.M. Moreno-Vega. Métodos multi-arranque. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:49–60, 2003.
- [MMB98] V. Maniezzo, A. Mingozzi, and R. Baldacci. A bionomic approach to the capacitated p-median problem. *Journal of Heuristics*, 4:263–280, 1998.
- [Mos99] P. Moscato. Memetic algorithms: A short introduction. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, volume 219–234. McGraw-Hill, 1999.
- [MS98] K. Marriot and P. J. Stuckey. *Constraint with Programming*. MIT press, 1998.
- [MTF03] P. Mills, E.P.K. Tsang, and J. Ford. Applying an extended guided local search on the quadratic assignment problem. *Annals of Operations Research*, 118:121–135, 2003.
- [OCS00] M.J. Oates, D.W. Corne, and G.D. Smith, editors. *Telecommunications Optimization: Heuristic and Adaptive Techniques*. Wiley, 2000.
- [OK96] I.H. Osman and J.P. Kelly, editors. *Meta-Heuristics. Theory and Applications*. Kluwer academic P., Norwell MA, 1996.
- [Osm93] I. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operation Research*, 41:421–451, 1993.
- [PG96] G. Pesant and M. Gendreau. A view of local search in constraint programming. In E.C. Freuder, editor, *Principles and Practice of Constraint programming LNCE 1118*, pages 353–366, Berlin, 1996. Springer-Verlag.
- [Pir94] M. Pirlot. General local search heuristics in combinatorial optimization: A tutorial. *Belgian J. of Operations Research, Statistics and Computer Science*, 32:7–67, 1994.
- [Pir96] M. Pirlot. General local search methods. *European Journal of Operational Research*, 92(3):493–511, 1996.

- [PM03] C. Cotta P. Moscato. A gentle introduction to memetic algorithms. In G. Kochenberger F. Glover, editor, *Handbook of Metaheuristics*, volume 105-144. Kluwer Academic Publishers, 2003.
- [PS01] G. Kendall P.Cowling and E. Soubeiga. A parameter-free hyperheuristic for scheduling a sales summit. pages 127–131, Porto, Portugal, 2001. 4th metaheuristics International Conference, MIC'2001.
- [RA00] M. Randall and D. Abramson. A general meta-heuristic based solver for combinatorial optimisation problems. Technical report, School of Informtaion Technology, Bond University, 2000.
- [Ree93] Colin R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publications, 1993.
- [Ree03] C.R. Reeves. Genetic algorithms. In F. Glover and G. Kochenberger, editors, *Handbook on MetaHeuristics*, chapter 3. 2003.
- [RG03] M. Resende and J.L. González-Velarde. Grasp: Procedimientos de búsqueda miopes aleatorizados and adaptativos. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 19:61–76, 2003.
- [RH01] C.C. Ribeiro and P. Hansen, editors. *Essays and Surveys in Metaheuristics*, volume 15. Kluwer, 2001.
- [RH02] K. E. Rosing and M. John Hodgson. Heuristic concentration for the p-median: an example demonstrating how and why it works. *Computers and Operations Research*, 29(10):1317–1330, 2002.
- [RM02] A. Roli and M. Milano. Magma: A multiagent architecture for metaheuristics. Technical report, DEIS Università degli Studi di Bonogna, 2002.
- [RPE] M. Resende, P. Pardalos, and S. Eksioglu. Parallel metaheuristics for combinatorial optimization <http://citeseer.nj.nec.com/resende99parallel.html>.
- [RR02] Colin R. Reeves and Jonathan E. Rowe, editors. *Genetic Algorithms - Principles and Perspectives. A Guide to GA Theory*, volume 20 of *Interfaces in Computer Science and Operations Research*. Kluwer Academic Publishers, 2002.
- [RR03] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G.G. Kochenberger, editors, *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.
- [RS95] V.J. Rayward-Smith. *Applications of Modern Heuristic Methods*. Alfred Waller, 1995.
- [RSORS96] V.J. Rayward-Smith, I.O. Osman, C.R. Reeves, and G.D. Smith, editors. *Modern Heuristic Search Methods*. Wiley, 1996.

- [Sal97] S. Salhi. A perturbation heuristic for a class of location problems. *Journal of the Operational Research Society*, 48:1233 – 1240, 1997.
- [SH96] T. Stützle and H. Hoos. Improvements on the ant systems: Introducing max-min ant system. Viena- Austria, 1996. Int. Conf. on Artif. Neural Networks and Genetic Alg.,.
- [SH97] T. Stützle and H. Hoos. Max-min ant system and local search for combinatorial optimization problems. 1997.
- [SLC99] A. Schaffer, M. Lenzerini, and M. Cadoli. Local++: a c++ framework for combinatorial search problems. Technical report, Dipt. di informatica e Sistemistica. Universita di Roma, Roma, Italy, 1999.
- [Stü99] T. Stützle. Local search algorithms for combinatorial problems-analysis, algorithms and news applications. DISKI Dissertationen zur Künstliken Intelligenz., 1999.
- [SVVW80] E.A. Silver, R. Victor, V. Vidal, and D. De Werra. A tutorial on heuristic methods. *European Journal of Operational Research*, 5:153–162, 1980.
- [TBM97] D. B. Fogel T. Ba“ck and Z. Michalewicz, editors. *Handbook of evolutionary computation*. Oxford University Press, 1997.
- [Teg01] J. Teghem. Solving multi-objctive combinatorial optimisation with metaheuristics. 2001.
- [Vid93] R.V.V. Vidal. Applied simulated annealing. 396, 1993.
- [VMOR99] S. Voss, S. Martello, I.H. Osman, and C. Roucairol, editors. *Meta-Heuristics: Trends in Local Search paradigms for Optimization*. Kluwer Academic, 1999.
- [Vos01] S. Voss. In Springer-Verlag, editor, *Local Search for plannng and Scheduling LNAI 2148*, pages 1–23, Berlin, 2001.
- [Vou97] C. Voudouris. *Guide Local Search for Combinatorial Optimisation Problems*. PhD thesis, Dept. of Computer Science. University of Essex, 1997.
- [VP00] José M. Moreno Vega and José A. Moreno Pérez. *Heurísticas en Optimización*. Gobierno de Canarias. Consejería de Educación Cultura y Deportes. Dirección General de Universidades e Investigación. Colección Textos Universitarios, 2000.
- [VT95] C. Voudouris and E.P. K. Tsang. Guide local search. Technical report, Dept. of Computer Science. University of Essex, 1995.
- [VT99] C. Voudouris and E.P.K. Tsang. Guided local search. *European Journal of Operational Research*, 113(2):469–499, 1999.
- [VT00] C. Voudouris and E.P.K. Tsang. Guide local search. <http://cswww.essex.ac.uk/Research/CSP/gls.html>, 2000.

- [VT03] C. Voudouris and E.P.K. Tsang. Guided local search. In F. Glover and G. Kochenberger, editors, *Handbook on MetaHeuristics*, chapter 7. 2003.
- [VW02] S. Voss and D.L. Woodruff, editors. *Optimization Software Class Libraries*, volume 18 of *OPERATIONS RESEARCH/COMPUTER SCIENCE INTERFACES*. Kluwer Academic Publishers, 2002.
- [Woo97] D.L. Woodruff. A class library for heuristic search optimization. *INFORMS Computer Science Technical Section*, 18 (2):1–5, 1997.
- [XJL⁺02] Z.-B. Xu, H.-D. Jin, K.-S. Leung, Y. Leung, and C.-K. Wong. An automata network for performing combinatorial optimization. *Neurocomputing*, 47:59–83, 2002.
- [YI01] M. Yagiura and T. Ibaraki. On metaheuristic algorithms for combinatorial optimization problems. *Systems and Computers in Japan*, 32(3):33–55, 2001.
- [YI02] M. Yagiura and T. Ibaraki. Local search. In P.M. Pardalos and M.G.C. Resende, editors, *Handbook of Applied Optimization*, pages 104–123. Oxford University Press, 2002.
- [ZEV89] S.H. Zanakis, J.R. Evans, and A.A. Vazacopoulos. Heuristic methods and applications: a categorized survey. *European Journal of Operational Research*, 43:88–110, 1989.