

# Una Visión General de los Algoritmos Meméticos

Carlos Cotta

Dept. Lenguajes y Ciencias de la Computación, ETSI Informática,  
University of Málaga, Campus de Teatinos, 29071 - Málaga, Spain.

ccottap@lcc.uma.es

## Resumen

Los algoritmos meméticos (MAs) constituyen un paradigma de optimización basado en la explotación sistemática de conocimiento acerca del problema que se desea resolver, y de la combinación de ideas tomadas de diferentes metaheurísticas, tanto basadas en población como basadas en búsqueda local. De la utilidad práctica de los MAs da cuenta la plétora de problemas de optimización complejos que han podido atacarse con éxito gracias a estas técnicas. En este trabajo se proporciona una visión general del paradigma de los MAs, describiendo la arquitectura básica de estas técnicas, y proporcionando algunas directrices para el diseño de un MA satisfactorio en un cierto dominio. También se incluye una panorámica de algunas de aplicaciones destacadas de los MAs, y algunas reflexiones en torno al presente y al futuro de estos métodos.

## 1. Introducción

Uno de los periodos clave en la historia de la Optimización como disciplina lo constituyen sin duda alguna las primeras décadas de la segunda mitad del siglo XX. En un tiempo en el que la concepción natural de la resolución de problemas era la obtención de la solución óptima al mismo (o cuanto menos de una solución con una garantía de aproximación al óptimo), empezó a tomar cuerpo una desagradable realidad: existían muchos problemas de utilidad cierta para los que no sólo resultaba insostenible plantear una resolución exacta, sino que ni tan siquiera un enfoque aproximado con garantías realistas era aceptable en la práctica. Esto abrió el camino a diferentes líneas de investigación para dar respuesta a esta problemática, y que más adelante desembocarían en lo que hoy se conoce como *metaheurísticas*. Entre las mismas, deben destacarse los algoritmos evolutivos [1, 2, 3, 4] (EAs<sup>1</sup>) por estar íntimamente relacionados con el tema que nos ocupa: los algoritmos meméticos.

A pesar de que estas técnicas fueran consideradas en su momento por parte de la comunidad científica como un “reconocimiento de la derrota”, el tiempo ha demostrado su utilidad como punta de lanza tecnológica en la optimización de problemas reales. Por supuesto, este éxito no es exclusivo de los EAs, sino que se extiende a otras técnicas relacionadas tales como el recocido simulado [5] (SA), la búsqueda tabú [6] (TS), etc. Parte de la justificación (y de hecho, del enfoque metodológico) de los algoritmos meméticos (MAs) se haya precisamente en el éxito de métodos de optimización tan diversos. En este sentido, debe considerarse que en ciertos campos se desarrolló una tendencia al purismo algorítmico, esto es, a no apreciar como característica esencial de estas técnicas su flexibilidad y capacidad de asimilación de elementos algorítmicos externos o ad hoc, que pudieran acercar la técnica de resolución al problema que resolver en cada momento. No fue hasta mediados de los noventa cuando la formulación del así denominado

---

<sup>1</sup>En este y en sucesivos acrónimos se empleará la versión inglesa por motivos de consistencia con la literatura, y para evitar posibles fuentes de confusión o ambigüedad.

Teorema de *No Free Lunch* por Wolpert and Macready [7] dio pie a una suerte de catarsis, a partir de la cual quedó definitivamente claro que un algoritmo de búsqueda u optimización se comporta en estricta concordancia con la cantidad y calidad del conocimiento específico del problema que incorpora. Mirando retrospectivamente a estos años nos encontramos con que esta filosofía que comenzó a imponerse de manera generalizada a finales del siglo XX ya estaba siendo promulgada de hecho con anterioridad por diversos investigadores, e.g., Hart and Belew [8], Davis [9], y Moscato [10]. El paradigma de los MAs surgiría precisamente a partir del trabajo de Pablo Moscato [11, 12, 13].

Los MAs son una familia de metaheurísticas que intentan aunar ideas y conceptos de diferentes técnicas de resolución, como por ejemplo EAs y TS. El adjetivo “memético” viene del término inglés *meme*, acuñado por R. Dawkins [14] para designar al análogo del gen en el contexto de la evolución cultural. Resulta conveniente resaltar sin embargo que el empleo de esta terminología no representa un propósito de adherirse a una metáfora de funcionamiento concreta (la evolución cultural en este caso), sino más bien lo contrario: hacer explícito que se difumina la inspiración puramente biológica, y se opta por modelos más genéricos en los que se manipula, se aprende y se transmite información. En relación con esto último y a la forma en la que más comúnmente un MA puede implementarse, pueden encontrarse diversos trabajos que hacen uso de nombre alternativos para referirse a éstos (e.g., EAs híbridos o lamarckianos), o que aun usando el propio término MA, hacen una interpretación muy restringida del mismo. Sea como fuere, puede decirse que un MA es una estrategia de búsqueda en la que una población de agentes optimizadores compiten y cooperan de manera sinérgica [10]. Más aún, estos agentes hacen uso explícito de conocimiento sobre el problema que se pretende resolver, tal como sugiere tanto la teoría como la práctica [15]. La siguiente sección proporciona una descripción algorítmica más detallada de los MAs.

## 2. Un Algoritmo Memético Básico

Los MAs son metaheurísticas basadas en población. Esto quiere decir que mantienen un conjunto de soluciones candidatas para el problema considerado. De acuerdo con la jerga empleada en EAs, cada una de estas soluciones tentativas es denominada un *individuo*. Tal como se anticipó anteriormente, la naturaleza de los MAs sugiere que el término *agente* es no obstante más apropiado. El motivo básico es el hecho de que “individuo” denota un ente *pasivo* que está sujeto a los procesos y reglas evolutivas, mientras que el término “agente” implica la existencia de un comportamiento *activo*, dirigido a propósito a la resolución de un cierto problema. Dicho comportamiento activo se ve reflejado en diferentes constituyentes típicos del algoritmo, como por ejemplo técnicas de búsqueda local.

La Figura 1 muestra el esquema general de un MA. Como en los EAs, la población de agentes está sujeta a los procesos de competición y cooperación mutua. Lo primero se consigue a través de los bien conocidos procedimientos de selección (línea 6) y reemplazo (línea 12): a partir de la información que proporciona una función de guía ad hoc se determina la bondad de los agentes en *pop*; acto seguido, se selecciona una parte de los mismos para pasar a la fase reproductiva atendiendo a dicha bondad. Posteriormente, se vuelve a hacer uso de esta información para determinar qué agentes serán eliminados de la población para hacer sitio a los nuevos agentes. En ambos casos –selección y reemplazo– pueden usarse cualesquiera de las estrategias típicas de los EAs, e.g., torneo, ranking, elitismo, etc.

En cuanto a la cooperación, ésta se consigue a través de la reproducción. En esta fase se crean nuevos agentes a partir de los existentes mediante el empleo de una serie de operadores de reproducción. Tal como se muestra en la Figura 1, líneas 7–11, pueden considerarse un número arbitrario  $\#op$  de tales operadores, que se aplican secuencialmente a la población de manera segmentada, dando lugar a varias poblaciones intermedias  $auxpop[i]$ ,  $0 \leq i \leq \#op$ ,

### Algoritmo Memético

ENTRADA: una instancia  $I$  de un problema  $P$ .

SALIDA: una solución  $sol$ .

```
// generar población inicial
1: para  $j \leftarrow 1:popsiz$ e hacer
2:   sea  $ind \leftarrow$  GenerarSoluciónHeuristica ( $I$ )
3:   sea  $pop[j] \leftarrow$  MejoraLocal ( $ind, I$ )
4: finpara
5: repetir // bucle generacional
   // Selección
6:   sea  $criadores \leftarrow$  SeleccionarDePoblación ( $pop$ )
   // Reproducción segmentada
7:   sea  $auxpop[0] \leftarrow pop$ 
8:   para  $j \leftarrow 1:\#op$  hacer
9:     sea  $auxpop[j] \leftarrow$  AplicarOperador ( $op[j], auxpop[j - 1], I$ )
10:  finpara
11:  sea  $newpop \leftarrow auxpop[\#op]$ 
   // Reemplazo
12:  sea  $pop \leftarrow$  ActualizarPoblación ( $pop, newpop$ )
   // Comprobar convergencia
13:  si Convergencia ( $pop$ ) entonces
14:    sea  $pop \leftarrow$  RefrescarPoblación ( $pop, I$ )
15:  finsi
16: hasta CriterioTerminación ( $pop, I$ )
17: devolver Mejor ( $pop, I$ )
```

Figura 1: Plantilla general de un MA

donde  $auxpop[0]$  está inicializada a  $pop$ , y  $auxpop[\#op]$  es la descendencia final. En la práctica, la situación más típica es la de utilizar simplemente tres operadores: recombinación, mutación, y mejora local. Apréciase en la línea 9 del pseudocódigo que estos operadores reciben no sólo las soluciones sobre las que actúan, sino también la instancia  $I$  que se desea resolver. Con esto se ilustra el hecho de que los operadores de un MA son conscientes del problema, y basan su funcionamiento en el conocimiento que incorporan sobre el mismo (a diferencia de los modelos más clásicos de EA).

Uno de los procesos reproductivos que mejor encapsula la cooperación entre agentes (dos, o más [16]) es la recombinación. Esto se consigue mediante la construcción de nuevas soluciones a partir de la información relevante contenida en los agentes cooperantes. Por “relevante” se entiende que los elementos de información considerados tienen importancia a la hora de determinar (en un sentido o en otro) la calidad de las soluciones. Ésta es sin duda una noción interesante que se aleja de las más clásicas manipulaciones sintácticas, típicas de EAs simples. Volveremos a esto más adelante, en la próxima sección.

El otro operador clásico –la mutación– cumple el rol de “mantener vivo el fuego”, inyectando nueva información en la población de manera continua (pero a ritmo bajo, ya que de lo contrario el algoritmo se degradaría a una pura búsqueda aleatoria). Por supuesto, esta interpretación es la que proviene del área de los algoritmos genéticos [17], y no necesariamente coincide con la de otros investigadores (aquellos del área de la programación evolutiva [1] sin ir más lejos). De hecho, en ocasiones se ha aducido que la recombinación no es más que una *macro-mutation*, y

ciertamente ese puede ser el caso en numerosas aplicaciones de los EAs en los que este operador de recombinación simplemente realiza una mezcla aleatoria de información. Sin embargo, no cabe hacer una apreciación similar en el campo de los MAs, ya que en éstos la recombinación se realiza típicamente mediante el empleo de estrategias *astutas*, y por lo tanto contribuyen de manera esencial a la búsqueda.

Finalmente, una de las características más distintivas de los MAs es el empleo de estrategias de búsqueda local (LS). Éstas (nótese que pueden emplearse diferentes estrategias de LS en diferentes puntos del algoritmo) constituyen una de las razones esenciales por las que es apropiado usar el término “agente” en este contexto: su funcionamiento es local, y en ocasiones incluso autónomo. De esta manera, un MA puede verse como una colección de agentes que realizan una exploración autónoma del espacio de búsqueda, cooperando en ocasiones a través de la recombinación, y compitiendo por recursos computacionales a través de los mecanismos de selección/reemplazo.

El pseudocódigo de la Figura 1 muestra un componente que merece asimismo atención: el procedimiento *RefrescarPoblación* (líneas 13–15). Este procedimiento tiene suma importancia con vistas al aprovechamiento de los recursos computacionales: si en un determinado instante de la ejecución todos los agentes tienen un estado similar (esto es, se ha producido convergencia), el avance de la búsqueda se torna muy complejo. Este tipo de circunstancias puede detectarse a través del empleo de medidas tales como la entropía de Shannon [18], fijando un umbral mínimo por debajo del cual se considera que la población ha degenerado. Obviamente, dicho umbral depende de la representación de problema que se esté usando, y debe decidirse por lo tanto de manera particular en cada caso.

### 3. Diseño de MAs Efectivos

Atacar un cierto problema de optimización con MAs requiere instanciar la plantilla genérica descrita anteriormente, empleando para ello conocimiento del problema. Dado que el diseño de un algoritmo de búsqueda efectivo es en general tan complejo como los propios problemas que se desean resolver, nos encontramos ante la tesitura de tener que emplear directrices heurísticas para abordar dicho problema de diseño. A continuación se consideraran algunas de estas directrices para algunos de los componentes esenciales de los MAs.

#### 3.1. Representación

El primer elemento que debe determinarse es la representación de las soluciones que se va a usar. Es importante en este punto aclarar que *representación* no debe entenderse como meramente *codificación*, algo para lo que lo relevante son consideraciones relativas a consumo de memoria, complejidad de manipulación, etc. Muy al contrario, la representación hace referencia a la formulación abstracta de las soluciones desde el punto de vista del algoritmo [19]. En este sentido, recuérdese la mención a *información relevante* que se hizo en la Sección 2. Dada una cierta representación de las soluciones, éstas pueden entenderse como compuestas de determinadas *unidades de información*; si los operadores que emplea el MA son conscientes del problema atacado, las unidades de información que identifiquen deben servir para determinar si una solución es buena/prometedora o no. La dinámica del sistema debe entonces tender a retener las unidades de información que lleven asociadas un efecto positivo, y a eliminar aquellas que tengan connotación negativa.

El siguiente ejemplo puede ayudar a ilustrar este aspecto de la representación. Considérese un problema definido sobre un espacio de soluciones compuesto de todas las permutaciones de  $n$  elementos; estas soluciones pueden entenderse como compuestas por diferentes tipos de información [20], e.g.,

- *posicional*, i.e., el elemento  $e$  aparece en la posición  $j$ .
- *precedencia*, i.e., el elemento  $e$  aparece antes/después que el elemento  $e'$ .
- *adyacencia*, i.e., el elemento  $e$  aparece junto al elemento  $e'$ .

La relevancia de cada tipo de información dependerá obviamente del problema que se desea resolver. Por ejemplo, la información de adyacencia es importante para el problema del viajante de comercio (TSP), pero no así la información posicional. Por otra parte, se ha comprobado que esta última sí es relevante en problemas de planificación de cadenas de montaje (*flowshop scheduling*) (FS) [21], siendo la información de adyacencia menos importante en este caso. Esto quiere decir que un operador de recombinación como ER (*edge-recombination*) [22] funcionará mejor que un operador basado en información posicional como PMX (*partially-mapped crossover*) [23] o UCX (*uniform cycle crossover*) [21] en el TSP, pero los últimos funcionarán mejor sobre FS.

No es sorprendente a la vista de lo anterior que la obtención de métodos o medidas para cuantificar la bondad de una cierta representación para un cierto problema haya sido y sea un tema de gran interés. Ha habido diferentes propuestas en este sentido: *epistasis* (i.e., la influencia no aditiva que sobre la función objetivo tiene la combinación de varias unidades de información) [24, 25], *varianza en la adecuación de formas* (i.e., varianza en los valores que devuelve la función objetivo para soluciones que comparten un cierto conjunto de unidades de información) [26], y *correlación de adecuación* (correlación entre los valores de la función objetivo para entre unas soluciones y sus descendientes directos) [27, 28]. Debe reseñarse que además de usar una métrica para predecir cuán bueno puede ser el rendimiento de un cierto operador pre-existente (i.e., *análisis inverso*), pueden definirse nuevos operadores ad hoc para manipular la mejor representación (*análisis directo*) [13].

Sea cual fuere la métrica usada para cuantificar la bondad de una representación concreta, hay otras consideraciones que también pueden jugar un papel determinante en el rendimiento final del algoritmo, tales como por ejemplo la existencia de restricciones en el espacio de búsqueda. Esta última problemática puede atacarse de tres maneras: (i) usando funciones de penalización que dirijan la búsqueda hacia regiones factibles, (ii) usando mecanismos de reparación que produzcan soluciones factibles a partir de soluciones infactibles, y (iii) usando operadores reproductivos que permanezcan siempre dentro de la zona factible. En los dos primeros casos es posible mantener la complejidad de la representación a un nivel más bajo (aunque lógicamente el algoritmo se beneficiará de cualquier conocimiento que adicional que pudiera usarse aquí). En el tercer caso, es responsabilidad de la representación o de los operadores el garantizar la factibilidad, y esto conllevará una complejidad adicional. Así, es posible definir representaciones indirectas que mediante el empleo de *decodificadores* garanticen la factibilidad de las soluciones representadas. La idea básica es utilizar un mecanismo sofisticado para pasar del genotipo al fenotipo, de manera que no sólo se consigan soluciones factibles, sino que además se introduzca conocimiento del problema que facilite que éstas sean de calidad (e.g., [29, 30, 31] entre otras).

### 3.2. Operadores Reproductivos

La generación de nuevas soluciones durante la fase reproductiva se realiza mediante la manipulación de las unidades de información relevantes que se han identificado. A tal fin, puede emplearse cualquiera de las plantillas genéricas definidas para ello, e.g., RRR (*random respectful recombination*), RAR (*random assorting recombination*), y RTR (*random transmitting recombination*) entre otras [32]. En cualquier caso, huelga decir que el rendimiento de algoritmo se verá beneficiado si en lugar de manipular las unidades de información a ciegas, se hace de manera inteligente empleando conocimiento del problema. Desde un punto de vista general, esta inclusión de conocimiento del problema en la manipulación de las unidades de información tiene dos

vertientes: la selección de las características parentales que serán transmitidas a la descendencia, y la selección de las características no-parentales que serán incluidas en la misma.

En relación a la selección de la información contenida en los padres que debe transmitirse a los hijos, la evidencia experimental aconseja conservar aquellas características comunes a ambos padres (e.g., [22, 33]). Una vez hecho esto, el descendiente puede completarse de diferentes maneras. Así, Radcliffe y Surry [26] proponen el empleo de estrategias de búsqueda local o de esquemas de enumeración implícitos. Estos últimos pueden usarse también para encontrar la mejor combinación posible de la información parental [34, 35, 36, 37] (dependiendo de las características de la representación, sería posible que esta combinación no necesariamente respetara las propiedades comunes). Puede apreciarse fácilmente que este tipo de recombinación sería monótono en el sentido de que los hijos serían siempre al menos tan buenos como los padres.

Hasta cierto punto podría hacerse un análisis similar del operador de mutación, si bien es verdad que éste juega un papel bien distinto: introducir nueva información en la población. En principio, esto puede conseguirse mediante la eliminación de ciertas unidades de información de una solución, y su substitución por información puramente aleatoria, o por información obtenida por alguno de los métodos de completado descritos anteriormente. Sin embargo, hay que resaltar que el papel de la mutación tiene ciertos matices diferenciadores en MAs frente a los clásicos EAs. De hecho, es posible incluso que un MA no tenga un operador de mutación diferenciado, sino que éste esté simplemente empotrado en la búsqueda local, e.g., véase [38, 39]. Uno de los motivos es el hecho de que los MAs están dotados de mecanismos de reinicio de la población (véase la Sección 3.4), y en ciertos contextos puede ser mejor dejar converger la población rápidamente y luego reiniciar, que diversificar constantemente la búsqueda. En cualquier caso, hay situaciones en las que mutación sí adquiere un papel determinante, y en las que incluso se emplean varios operadores de mutación. Esto se realiza bien por el empleo de diferentes vecindades (e.g., [40]), o definiendo mutaciones débiles y fuertes que introduzcan diferentes niveles de perturbación (e.g., [41]). Nótese que en cierto sentido el empleo de diferentes operadores reproductivos implica de manera implícita la consideración de diferentes representaciones y/o vecindades durante la ejecución, muy en la línea de los que se hace en la búsqueda en vecindades variables [42] (VNS).

Es posible introducir también conocimiento del problema mediante el empleo de heurísticas constructivas en los operadores de inicialización usados para la generación de la población inicial (Figura 1, línea 2). Por ejemplo, se han empleado estrategias voraces para este propósito en [43, 44].

### 3.3. Búsqueda Local

La presencia de componentes de búsqueda local (LS) es –tal como se comento anteriormente– una de las características más distintivas de los MAs. El hecho de que la mayoría de los MAs incorporen LS es una de las causas por las que a veces se pueden encontrar simplificaciones del tipo  $MA = EA + LS$ , y que deben evitarse; véase [11, 12, 13] para más detalles. De hecho, es posible encontrar enfoques metaheurísticos con muy similar filosofía a la de los MAs, y que sin embargo no pueden llamarse evolutivos a no ser que se asuma una definición tan amplia del termino que prácticamente abarque a cualquier método basado en población. La técnica de búsqueda dispersa (SS) [45] es un buen ejemplo en este sentido. Por otra parte, no es extraño encontrar enfoques evolutivos en los que el conocimiento del problema se concentra más en el operador de recombinación que en el uso de una búsqueda local, e.g., [35, 46]. En cualquier caso, esta claro que  $EA + LS \subset MA$ , y que el componente LS es típicamente uno de los que más contribuyen al éxito del algoritmo.

Las técnicas de mejorar local pueden modelarse como trayectorias en el espacio de búsqueda tal que soluciones vecinas en dicha trayectoria difieren en una pequeña cantidad de unidades de información. Esta definición idealizada puede requerir no obstante diferentes matizaciones si

por ejemplo se emplea TS para este fin, e.g., [47, 36, 48] entre otras muchas. Así, es normal que muchas implementaciones de TS usen estrategias de intensificación que hagan que en ciertos momentos la búsqueda se continúe por ciertas soluciones anteriores de calidad (así, más que a un camino lineal, el recorrido a través del espacio de búsqueda se asemejaría a una trayectoria ramificada). Más aún tanto en TS como en otras metaheurísticas tales como SA, puede darse que la calidad de las soluciones no se incremente de manera monótona, sino que en ciertos momentos empeore con la finalidad de poder escapar de óptimos locales. Por supuesto, al final de la ejecución del procedimiento se conserva la mejor solución encontrada, y no la última generada.

A la hora de implementar el componente LS es importante determinar el criterio de terminación. Si se está empleando una técnica simple de escalada (HC) puede tener sentido determinar si la solución actual es un óptimo local y detener el procedimiento únicamente en ese caso. Obviamente, esto no es posible si se usa TS o SA, ya que estas técnicas tienen capacidades globales de optimización, por lo que lo más común es definir un tope computacional máximo (e.g., en forma de número de soluciones exploradas). Lógicamente, en este caso la solución final no tiene por qué ser un óptimo local (como algunas descripciones erróneas de MAs aseguran). Además, debe encontrarse un equilibrio adecuado entre el esfuerzo computacional que se realiza durante LS y el que la búsqueda poblacional subyacente realiza. La importancia de este hecho ha dado lugar a la noción de *lamarckismo parcial* [49, 50], esto es, no usar siempre la búsqueda local, sino únicamente sobre algunas soluciones o bien seleccionadas aleatoriamente, o bien en función de su calidad, o bien según algún otro método (véase también [51]).

Del mismo modo que se pueden definir métricas para cuantificar la bondad de una representación (u operador que trabaje sobre la misma), pueden definirse métricas que ayuden a predecir si una determinada definición de vecindad puede ser beneficiosa. Por ejemplo, la correlación entre *distancia* y adecuación [52, 53] (FDC) es una de las propuestas. Esencialmente, la distancia mencionada se entiende como el número de movimientos (saltos de vecindad) que hay que realizar para pasar de un óptimo local al óptimo global. Si el coeficiente de correlación entre esta distancia y la calidad de la función objetivo es alta, entonces la calidad de las soluciones tiende a mejorar al acercarse al óptimo global, y la dinámica evolutiva del MA lo llevará a su cercanía. Si la correlación fuera negativa, el problema sería engañoso para el MA, ya que los óptimos locales mejores se alejarían del óptimo global.

Otro aspecto importante en relación al paisaje de búsqueda es su topología global, y más precisamente si la relación de vecindad es regular o no, y que relación guarda con la calidad de las soluciones. Bierwirth *et al.* [54] han estudiado esta circunstancia para un problema de planificación, y han encontrado que las mejores soluciones tienen una mayor conectividad, lo que las hace más fácilmente alcanzables, incluso por mor de pura deriva genética. No todos los problemas tienen esta propiedad, y de hecho, Cotta y Fernández han encontrado que la representación directa para la búsqueda de reglas de Golomb de tamaño mínimo tiene precisamente la propiedad opuesta [55].

### 3.4. Gestión de la Diversidad

Hay diferentes maneras de enfocar la diversidad en algoritmos basados en población. Por un lado pueden considerarse métodos de preservación de la diversidad, dentro de los cuales se engloban claramente los operadores de mutación. Éstos no son los únicos mecanismos posibles sin embargo. Por ejemplo, pueden introducirse soluciones completamente nuevas (los así llamados “inmigrantes aleatorios”) [56] en la población, e.g., [57], o pueden emplearse poblaciones con estructura espacial [58]. En este último caso, se restringe el emparejamiento de agentes o el reemplazo de los mismos a elementos situados en posiciones vecinas dentro de la estructura topológica de la población. Esto causa un ralentizamiento de la propagación de información a

través de la población, con lo que se impide (o al menos dificulta) que algunos *super-agentes* tomen rápidamente control de la misma y destruyan toda diversidad.

En la literatura se han propuesto diferentes topologías para organizar la población, i.e., anillos, rejillas, hipercubos, etc. En relación con los MAs, una de las opciones más exitosas ha sido una estructura jerárquica en forma de árbol ternario [59, 41, 60, 61]. Esta topología se ha combinado con una estrategia para organizar la distribución de las soluciones en función de su calidad. Más concretamente, cada nodo del árbol está restringido a tener una solución mejor que cualquiera de los nodos descendientes. Esto implica que cuando un agente tiene una solución mejor que la de su antecesor directo en el árbol, las intercambian. De esta manera, hay un continuo flujo de soluciones de calidad hacia la parte superior del árbol, lo cual también garantiza que cuando se realiza una recombinación, las soluciones que toman parte en ella son de calidad similar.

Como complemento a los mecanismos de preservación anteriores se pueden considerar también los mecanismos de restauración de la diversidad: cuando se detecta que la diversidad ha caído por debajo de un cierto umbral, o cuando la dinámica del algoritmo apunta a un estado de degeneración en la búsqueda [62] se activa uno de estos mecanismos para relanzarla. Una posibilidad en este sentido es emplear hipermutación [63, 41] (cf. mutación pesada, véase Sección 3.2). Alternativamente, la población puede refrescarse mediante la llegada masiva de inmigrantes aleatorios que sustituyan a toda la población, salvo a algunas soluciones de elite.

## 4. Aplicaciones de los MA

Uno de los campos más fructíferos para los MAs es el ámbito de la optimización combinatoria, para el que estas técnicas cuentan con cientos de aplicaciones. Eso no es sorprendente si tenemos en consideración que existen miles de problemas de optimización pertenecientes a la clase NP, donde los MA se han mostrado de gran valor. De entre todas éstas, y a modo ilustrativo, pueden destacarse las siguientes: problemas de particionado en grafos [64, 65], partición de números [66, 59], conjunto independiente de cardinalidad máxima [67, 68], empaquetado [69], coloreado de grafos [70, 71], recubrimiento de conjuntos [72], planificación de tareas en una máquina con tiempos de “set-up” y fechas de entrega [73, 74], planificación de tareas en varias máquinas [75, 76], problemas de asignación generalizados [77], problemas de mochila multidimensional [78, 79], programación entera no-lineal [80], asignación cuadrática [81, 53], particionado de conjuntos [82], y muy especialmente el *problema del viajante de comercio* [83, 53, 84]. Es de destacar que en una gran parte de estas publicaciones los propios autores destacan que la metodología constituye el estado del arte para el problema en consideración, lo que es de interés debido a que estos son problemas “clásicos” en el área de la optimización combinatoria.

El paradigma fue utilizado en otros problemas menos conocidos, pero sin duda de igual importancia, como son: emparejamiento parcial de formas geométricas [85], optimización en “paisajes NK” [86], diseño de trayectorias óptimas para naves espaciales [87], asignación de frecuencias [88], construcción de árboles de expansión mínimos con restricciones de grado [89], problemas de emplazamiento [90, 91], optimización de rutas [92], problemas de transporte [93, 94], isomorfismos en grafos [95], problemas de biconexión de vértices [96], agrupamiento [97], telecomunicaciones [98], búsqueda de regleros de Golomb mínimos [99, 100], búsqueda de patrones estables en autómatas celulares [36, 101], identificación de sistemas no-lineales [102], programación de tareas de mantenimiento [103, 104], *open shop scheduling* [105, 40], *flowshop scheduling* [106, 44], planificación de proyectos [107, 108], planificación de almacén [109], planificación de producción [110, 111], confección de horarios [112, 113], planificación de turnos [114, 115], planificación de juegos deportivos [116] y planificación de exámenes [117, 118].

Los MAs, también han sido citados en la literatura de aprendizaje en máquinas y robótica como algoritmos genéticos híbridos. Destacamos algunas aplicaciones como por ejemplo: entre-



namiento de redes neuronales [119, 120], reconocimiento de características [121], clasificación de características [122, 123], análisis de series temporales [124], aprendizaje de comportamientos reactivos en agentes móviles [125], planificación de trayectorias [126, 127], control óptimo [128], etc.

En las áreas de la Electrónica y la Ingeniería podemos destacar: proyectos de VLSI [129], optimización de estructuras [130] y mecánica de fracturas [131], modelado de sistemas [132], control de reactores químicos [133], calibración de motores [134], problemas de diseño óptimo en Aeronáutica [135, 136], diseño de sistemas ópticos [137], control de tráfico [138], y planificación en problemas de potencia [139] entre otros.

Otras aplicaciones de estas técnicas pueden encontrarse en: Medicina [140, 141], Economía [142, 143], Oceanografía [144], Matemáticas [145, 146, 147], Procesamiento de imágenes y de voz [148, 149, 150], y un sinnúmero de ellas en Bioinformática ([49, 60, 151, 152] entre otras muchas).

## 5. Conclusiones

A diferencia de otras técnicas de optimización, los MAs fueron explícitamente concebidos como un paradigma ecléctico y pragmático, abierto a la integración de otras técnicas (metaheurísticas o no). En última instancia, esta habilidad para combinar de manera sinérgica diferentes métodos es una de las razones de su éxito. Los MAs proporcionan un marco de trabajo apropiado para integrar en un único motor de búsqueda diferentes heurísticas provechosas. En este sentido, los MAs deben considerarse no como competidores, sino como integradores: allá donde una metaheurística *pura* empiece a alcanzar sus límites, los MAs constituyen el siguiente paso natural.

Aunque existe un importante componente experimental en el diseño de los MAs, no por ello puede afirmarse que el paradigma se reduce a combinar varias técnicas y realizar pruebas experimentales para comprobar si es satisfactoria. Muy al contrario, todo el corpus teórico disponible tanto para técnicas basadas en población como para técnicas de búsqueda local es de aplicación en el diseño de un MA. Otras estrategias de gran interés en este área son el diseño por analogía, y el máximo aprovechamiento de los recursos computacionales. Téngase en cuenta en relación a esto último que una técnica de búsqueda local muy sofisticada puede proporcionar mejores resultados que un simple HC, pero necesitar mucho más tiempo de cómputo para ello. En problemas en los que el coste de evaluar una solución es grande, o en los que los tamaños de las vecindades son considerables, éste es un problema que debe tenerse muy en cuenta.

Está claro asimismo que nuestro mundo se está haciendo cada vez más complejo a un ritmo acelerado, al menos desde un punto de vista tecnológico. Los años venideros depararán nuevos desafíos desde el punto de vista de la optimización a los que habrá que dar respuesta con metaheurísticas. No sólo habrá que hacer frente a problemas de optimización a gran escala, sino que estos mismos serán cada vez más complejos per se. Para ello, las técnicas de optimización tendrán que adaptarse a esta complejidad, dejando de lado los tradicionales enfoques unidimensionales y puramente secuenciales. Así, algunos de los aspectos de los MAs que tomarán cada vez más relevancia son la optimización multi-objetivo [51, 153, 154], la auto-adaptación [155, 156], y el funcionamiento autónomo [157, 158]. Como puede apreciarse, ya hay algunas propuestas en este sentido, siendo posible además aprovechar ideas de técnicas relacionadas tales como las hiperheurísticas [159, 160]. Otros métodos jugarán también un papel esencial, e.g., las técnicas de reducción a un kernel seguro comúnmente empleadas dentro del campo de la complejidad parametrizada [161]. Dado que el eclecticismo es esencial para adaptarse a este nuevo escenario, sólo cabe decir que el futuro es prometedor para los MAs.

## Agradecimientos

El autor agradece el apoyo parcial del MCyT a través del contrato TIN2005-08818-C04-01.

## Referencias

- [1] Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence through Simulated Evolution. John Wiley & Sons, New York (1966)
- [2] Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
- [3] Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart (1973)
- [4] Schwefel, H.P.: Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik. Diplomarbeit, Technische Universität Berlin, Hermann Föttinger-Institut für Strömungstechnik (1965)
- [5] Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220** (1983) 671–680
- [6] Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Boston, MA (1997)
- [7] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1(1)** (1997) 67–82
- [8] Hart, W.E., Belew, R.K.: Optimizing an arbitrary function is hard for the genetic algorithm. In Belew, R.K., Booker, L.B., eds.: Proceedings of the 4<sup>th</sup> International Conference on Genetic Algorithms, San Mateo CA, Morgan Kaufmann (1991) 190–195
- [9] Davis, L.D.: Handbook of Genetic Algorithms. Van Nostrand Reinhold Computer Library, New York (1991)
- [10] Moscato, P.: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA (1989)
- [11] Moscato, P.: Memetic algorithms: A short introduction. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. McGraw-Hill, Maidenhead, Berkshire, England, UK (1999) 219–234
- [12] Moscato, P., Cotta, C.: A gentle introduction to memetic algorithms. In Glover, F., Kochenberger, G., eds.: Handbook of Metaheuristics. Kluwer Academic Publishers, Boston MA (2003) 105–144
- [13] Moscato, P., Cotta, C., Mendes, A.S.: Memetic algorithms. In Onwubolu, G.C., Babu, B.V., eds.: New Optimization Techniques in Engineering. Springer-Verlag, Berlin Heidelberg (2004) 53–85
- [14] Dawkins, R.: The Selfish Gene. Clarendon Press, Oxford (1976)
- [15] Culberson, J.: On the futility of blind search: An algorithmic view of “No Free Lunch”. *Evolutionary Computation* **6** (1998) 109–127

- [16] Eiben, A.E., Raue, P.E., Ruttkay, Z.: Genetic algorithms with multi-parent recombination. In Davidor, Y., Schwefel, H.P., Männer, R., eds.: *Parallel Problem Solving From Nature III*. Volume 866 of *Lecture Notes in Computer Science*. Springer-Verlag (1994) 78–87
- [17] Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA (1989)
- [18] Davidor, Y., Ben-Kiki, O.: The interplay among the genetic algorithm operators: Information theory tools used in a holistic way. In Männer, R., Manderick, B., eds.: *Parallel Problem Solving From Nature II*, Amsterdam, Elsevier Science Publishers B.V. (1992) 75–84
- [19] Radcliffe, N.J.: Non-linear genetic representations. In Männer, R., Manderick, B., eds.: *Parallel Problem Solving From Nature II*, Amsterdam, Elsevier Science Publishers B.V. (1992) 259–268
- [20] Fox, B.R., McMahon, M.B.: Genetic operators for sequencing problems. In Rawlins, G.J.E., ed.: *Foundations of Genetic Algorithms I*, San Mateo, CA, Morgan Kaufmann (1991) 284–300
- [21] Cotta, C., Troya, J.M.: Genetic forma recombination in permutation flowshop problems. *Evolutionary Computation* **6** (1998) 25–44
- [22] Mathias, K., Whitley, L.D.: Genetic operators, the fitness landscape and the traveling salesman problem. In Männer, R., Manderick, B., eds.: *Parallel Problem Solving From Nature II*, Amsterdam, Elsevier Science Publishers B.V. (1992) 221–230
- [23] Goldberg, D.E., Lingle Jr., R.: Alleles, loci and the traveling salesman problem. In Grefenstette, J.J., ed.: *Proceedings of the 1<sup>st</sup> International Conference on Genetic Algorithms*, Hillsdale NJ, Lawrence Erlbaum Associates (1985) 154–159
- [24] Davidor, Y.: Epistasis Variance: Suitability of a Representation to Genetic Algorithms. *Complex Systems* **4** (1990) 369–383
- [25] Davidor, Y.: Epistasis variance: A viewpoint on GA-hardness. In Rawlins, G.J.E., ed.: *Foundations of Genetic Algorithms I*, San Mateo, CA, Morgan Kaufmann (1991) 23–35
- [26] Radcliffe, N.J., Surry, P.D.: Fitness Variance of Formae and Performance Prediction. In Whitley, L.D., Vose, M.D., eds.: *Foundations of Genetic Algorithms III*, San Francisco, CA, Morgan Kaufmann (1994) 51–72
- [27] Dzuber, J., Whitley, L.D.: Advanced Correlation Analysis of Operators for the Traveling Salesman Problem. In Schwefel, H.P., Männer, R., eds.: *Parallel Problem Solving from Nature III*. Volume 866 of *Lecture Notes in Computer Science*., Dortmund, Germany, Springer-Verlag, Berlin, Germany (1994) 68–77
- [28] Manderick, B., de Weger, M., Spiessens, P.: The Genetic Algorithm and the Structure of the Fitness Landscape. In Belew, R.K., Booker, L.B., eds.: *Proceedings of the 4<sup>th</sup> International Conference on Genetic Algorithms*, San Mateo, CA, Morgan Kaufmann (1991) 143–150
- [29] Aickelin, U., Dowsland, K.: Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of Scheduling* **3** (2000) 139–153

- [30] Cotta, C., Troya, J.: A hybrid genetic algorithm for the 0-1 multiple knapsack problem. In Smith, G., Steele, N., Albrecht, R., eds.: *Artificial Neural Nets and Genetic Algorithms 3*, Wien New York, Springer-Verlag (1998) 251–255
- [31] Varela, R., Puente, J., Vela, C.R.: Some issues in chromosome codification for scheduling with genetic algorithms. In Castillo, L., Borrajo, D., Salido, M.A., Oddi, A., eds.: *Planning, Scheduling and Constraint Satisfaction: From Theory to Practice*. Volume 117 of *Frontiers in Artificial Intelligence and Applications*. IOS Press (2005) 1–10
- [32] Radcliffe, N.J.: The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence* **10** (1994) 339–384
- [33] Oğuz, C., Ercan, M.F.: A genetic algorithm for hybrid flow-shop scheduling with multi-processor tasks. *Journal of Scheduling* **8** (2005) 323–351
- [34] Cotta, C., Aldana, J., Nebro, A., Troya, J.: Hybridizing genetic algorithms with branch and bound techniques for the resolution of the TSP. In Pearson, D., Steele, N., Albrecht, R., eds.: *Artificial Neural Nets and Genetic Algorithms 2*, Wien New York, Springer-Verlag (1995) 277–280
- [35] Cotta, C., Troya, J.M.: Embedding branch and bound within evolutionary algorithms. *Applied Intelligence* **18** (2003) 137–153
- [36] Gallardo, J.E., Cotta, C., Fernández, A.J.: A memetic algorithm with bucket elimination for the still life problem. In Gottlieb, J., Raidl, G., eds.: *Evolutionary Computation in Combinatorial Optimization*. Volume 3906 of *Lecture Notes in Computer Science.*, Berlin Heidelberg, Springer-Verlag (2006) 73–85
- [37] Ibaraki, T.: Combination with dynamic programming. In Bäck, T., Fogel, D., Michalewicz, Z., eds.: *Handbook of Evolutionary Computation*. Oxford University Press, New York NY (1997) D3.4:1–2
- [38] Maheswaran, R., Ponnambalam, S.G., Aranvidan, C.: A meta-heuristic approach to single machine scheduling problems. *International Journal of Advanced Manufacturing Technology* **25** (2005) 772–776
- [39] Wang, L., Zheng, D.Z.: A modified genetic algorithm for job-shop scheduling. *International Journal of Advanced Manufacturing Technology* **20** (2002) 72–76
- [40] Liaw, C.F.: A hybrid genetic algorithm for the open shop scheduling problem. *European Journal of Operational Research* **124** (2000) 28–42
- [41] França, P.M., Gupta, J., Mendes, A.S., Moscato, P., Veltink, K.J.: Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers and Industrial Engineering* **48** (2005) 491–506
- [42] Hansen, P., Mladenović, N.: Variable neighborhood search: Principles and applications. *European Journal of Operational Research* **130** (2001) 449–467
- [43] Cotta, C.: Scatter search with path relinking for phylogenetic inference. *European Journal of Operational Research* **169** (2006) 520–532
- [44] Yeh, W.C.: A memetic algorithm for the  $n/2$ /Flowshop/ $\alpha F + \beta C_{max}$  scheduling problem. *International Journal of Advanced Manufacturing Technology* **20** (2002) 464–473

- [45] Laguna, M., Martí, R.: Scatter Search. Methodology and Implementations in C. Kluwer Academic Publishers, Boston MA (2003)
- [46] Nagata, Y., Kobayashi, S.: Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. In Bäck, T., ed.: Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, EE.UU., San Mateo, CA, Morgan Kaufmann (1997) 450–457
- [47] Burke, E.K., Cowling, P.I., De Causmaecker, P., van den Berghe, G.: A memetic approach to the nurse rostering problem. Applied Intelligence **15** (2001) 199–214
- [48] Yamada, T., Reeves, C.R.: Solving the  $C_{sum}$  permutation flowshop scheduling problem by genetic local search. In: 1998 IEEE International Conference on Evolutionary Computation, Piscataway, NJ, IEEE Press (1998) 230–234
- [49] Cotta, C.: Memetic algorithms with partial lamarckism for the shortest common super-sequence problem. In Mira, J., Álvarez, J., eds.: Artificial Intelligence and Knowledge Engineering Applications: a Bioinspired Approach. Volume 3562 of Lecture Notes in Computer Science., Berlin Heidelberg, Springer-Verlag (2005) 84–91
- [50] Houck, C., Joines, J.A., Kay, M.G., Wilson, J.R.: Empirical investigation of the benefits of partial lamarckianism. Evolutionary Computation **5** (1997) 31–60
- [51] Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation **7** (2003) 204–223
- [52] Jones, T.C., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In Eshelman, L.J., ed.: Proceedings of the 6<sup>th</sup> International Conference on Genetic Algorithms, Morgan Kaufmann (1995) 184–192
- [53] Merz, P., Freisleben, B.: Fitness landscapes and memetic algorithm design. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. McGraw-Hill, Maidenhead, Berkshire, England, UK (1999) 245–260
- [54] Bierwirth, C., Mattfeld, D.C., Watson, J.P.: Landscape regularity and random walks for the job shop scheduling problem. In Gottlieb, J., Raidl, G.R., eds.: Evolutionary Computation in Combinatorial Optimization. Volume 3004 of Lecture Notes in Computer Science., Berlin, Springer-Verlag (2004) 21–30
- [55] Cotta, C., Fernández, A.: Analyzing fitness landscapes for the optimal golomb ruler problem. In Gottlieb, J., Raidl, G., eds.: Evolutionary Computation in Combinatorial Optimization. Volume 3248 of Lecture Notes in Computer Science., Berlin, Springer-Verlag (2005) 68–79
- [56] Grefenstette, J.J.: Genetic algorithms for changing environments. In Männer, R., Manderick, B., eds.: Parallel Problem Solving from Nature II, Amsterdam, North-Holland Elsevier (1992) 137–144
- [57] Hadj-Alouane, A.B., Bean, J.C., Murty, K.G.: A hybrid genetic/optimization algorithm for a task allocation problem. Journal of Scheduling **2** (1999) 181–201
- [58] Tomassini, M.: Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time. Springer-Verlag (2005)

- [59] Berretta, R., Cotta, C., Moscato, P.: Enhancing the performance of memetic algorithms by using a matching-based recombination algorithm: Results on the number partitioning problem. In Resende, M., Pinho de Sousa, J., eds.: *Metaheuristics: Computer-Decision Making*. Kluwer Academic Publishers, Boston MA (2003) 65–90
- [60] Mendes, A., Cotta, C., Garcia, V., França, P., Moscato, P.: Gene ordering in microarray data using parallel memetic algorithms. In Skie, T., Yang, C.S., eds.: *Proceedings of the 2005 International Conference on Parallel Processing Workshops*, Oslo, Norway, IEEE Press (2005) 604–611
- [61] Moscato, P., Mendes, A., Cotta, C.: Scheduling and production & control. In Onwubolu, G.C., Babu, B.V., eds.: *New Optimization Techniques in Engineering*. Springer-Verlag, Berlin Heidelberg (2004) 655–680
- [62] Cotta, C., Alba, E., Troya, J.M.: Stochastic reverse hillclimbing and iterated local search. In: *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington D.C., IEEE Neural Network Council - Evolutionary Programming Society - Institution of Electrical Engineers (1999) 1558–1565
- [63] Cobb, H.G.: An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. Technical Report AIC-90-001, Naval Research Laboratory, Washington DC (1990)
- [64] Merz, P., Freisleben, B.: Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning. *Evolutionary Computation* **8** (2000) 61–91
- [65] Yeh, W.C.: A memetic algorithm for the min  $k$ -cut problem. *Control and Intelligent Systems* **28** (2000) 47–55
- [66] Berretta, R., Moscato, P.: The number partitioning problem: An open challenge for evolutionary computation ? In Corne, D., Dorigo, M., Glover, F., eds.: *New Ideas in Optimization*. McGraw-Hill, Maidenhead, Berkshire, England, UK (1999) 261–278
- [67] Aggarwal, C., Orlin, J., Tai, R.: Optimized crossover for the independent set problem. *Operations Research* **45** (1997) 226–234
- [68] Hifi, M.: A genetic algorithm-based heuristic for solving the weighted maximum independent set and some equivalent problems. *Journal of the Operational Research Society* **48** (1997) 612–622
- [69] Reeves, C.: Hybrid genetic algorithms for bin-packing and related problems. *Annals of Operations Research* **63** (1996) 371–396
- [70] Coll, P., Durán, G., Moscato, P.: On worst-case and comparative analysis as design principles for efficient recombination operators: A graph coloring case study. In Corne, D., Dorigo, M., Glover, F., eds.: *New Ideas in Optimization*. McGraw-Hill, Maidenhead, Berkshire, England, UK (1999) 279–294
- [71] Dorne, R., Hao, J.: A new genetic local search algorithm for graph coloring. In Eiben, A., Bäck, T., Schoenauer, M., Schwefel, H.P., eds.: *Parallel Problem Solving From Nature V*. Volume 1498 of *Lecture Notes in Computer Science.*, Berlin, Springer-Verlag (1998) 745–754
- [72] Beasley, J., Chu, P.: A genetic algorithm for the set covering problem. *European Journal of Operational Research* **94** (1996) 393–404

- [73] França, P., Mendes, A., Moscato, P.: Memetic algorithms to minimize tardiness on a single machine with sequence-dependent setup times. In: Proceedings of the 5th International Conference of the Decision Sciences Institute, Athens, Greece, Atlanta, GA, USA, Decision Sciences Institute (1999) 1708–1710
- [74] Miller, D., Chen, H., Matson, J., Liu, Q.: A hybrid genetic algorithm for the single machine scheduling problem. *Journal of Heuristics* **5** (1999) 437–454
- [75] Mendes, A., Muller, F., França, P., Moscato, P.: Comparing meta-heuristic approaches for parallel machine scheduling problems with sequence-dependent setup times. In: Proceedings of the 15th International Conference on CAD/CAM Robotics & Factories of the Future, Aguas de Lindoia, Brasil. Volume 1., Campinas, SP, Brazil, Technological Center for Informatics Foundation (1999) 1–6
- [76] Min, L., Cheng, W.: Identical parallel machine scheduling problem for minimizing the makespan using genetic algorithm combined with simulated annealing. *Chinese Journal of Electronics* **7** (1998) 317–321
- [77] Chu, P., Beasley, J.: A genetic algorithm for the generalised assignment problem. *Computers & Operations Research* **24** (1997) 17–23
- [78] Beasley, J., Chu, P.: A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics* **4** (1998) 63–86
- [79] Gottlieb, J.: Permutation-based evolutionary algorithms for multidimensional knapsack problems. In Carroll, J., Damiani, E., Haddad, H., Oppenheim, D., eds.: ACM Symposium on Applied Computing 2000, ACM Press (2000) 408–414
- [80] Taguchi, T., Yokota, T., Gen, M.: Reliability optimal design problem with interval coefficients using hybrid genetic algorithms. *Computers & Industrial Engineering* **35** (1998) 373–376
- [81] Merz, P., Freisleben, B.: A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem. In Angeline, P., ed.: 1999 Congress on Evolutionary Computation (CEC'99), Piscataway, NJ, USA, IEEE Press (1999) 2063–2070
- [82] Levine, D.: A parallel genetic algorithm for the set partitioning problem. In Osman, I., Kelly, J., eds.: *Meta-Heuristics: Theory & Applications*. Kluwer Academic Publishers, Boston, MA, USA (1996) 23–35
- [83] Holstein, D., Moscato, P.: Memetic algorithms using guided local search: A case study. In Corne, D., Dorigo, M., Glover, F., eds.: *New Ideas in Optimization*. McGraw-Hill, Maidenhead, Berkshire, England, UK (1999) 235–244
- [84] Merz, P.: A comparison of memetic recombination operators for the traveling salesman problem. In Langdon, W., et al., eds.: *GECCO*, Morgan Kaufmann (2002) 472–479
- [85] Ozcan, E., Mohan, C.: Steady state memetic algorithm for partial shape matching. In Porto, V., Saravanan, N., Waagen, D., eds.: *Evolutionary Programming VII*. Volume 1447 of *Lecture Notes in Computer Science*., Springer, Berlin (1998) 527–536
- [86] Merz, P., Freisleben, B.: On the Effectiveness of Evolutionary Search in High-Dimensional *NK*-Landscapes. In Fogel, D., ed.: *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, Piscataway, NJ, USA, IEEE Press (1998) 741–745

- [87] Crain, T., Bishop, R., Fowler, W., Rock, K.: Optimal interplanetary trajectory design via hybrid genetic algorithm/recursive quadratic program search. In: Ninth AAS/AIAA Space Flight Mechanics Meeting, Breckenridge CO (1999) 99–133
- [88] Kassotakis, I., Markaki, M., Vasilakos, A.: A hybrid genetic approach for channel reuse in multiple access telecommunication networks. *IEEE Journal on Selected Areas in Communications* **18** (2000) 234–243
- [89] Raidl, G., Julstron, B.: A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem. In Carroll, J., Damiani, E., Haddad, H., Oppenheim, D., eds.: *ACM Symposium on Applied Computing 2000*, ACM Press (2000) 440–445
- [90] Hopper, E., Turton, B.: A genetic algorithm for a 2d industrial packing problem. *Computers & Industrial Engineering* **37** (1999) 375–378
- [91] Krzanowski, R., Raper, J.: Hybrid genetic algorithm for transmitter location in wireless networks. *Computers, Environment and Urban Systems* **23** (1999) 359–382
- [92] Rodrigues, A., Ferreira, J.S.: Solving the rural postman problem by memetic algorithms. In de Sousa, J.P., ed.: *Proceedings of the 4th Metaheuristic International Conference (MIC'2001)*, Porto, Portugal, July 16-20, 2001. (2001) 679–684
- [93] Gen, M., Ida, K., Yinzheng, L.: Bicriteria transportation problem by hybrid genetic algorithm. *Computers & Industrial Engineering* **35** (1998) 363–366
- [94] Novaes, A., De-Cursi, J., Graciolli, O.: A continuous approach to the design of physical distribution systems. *Computers & Operations Research* **27** (2000) 877–893
- [95] Torres-Velazquez, R., Estivill-Castro, V.: A memetic algorithm instantiated with selection sort consistently finds global optima for the error-correcting graph isomorphism. In Yao, X., ed.: *Proceedings of the IEEE 2002 Congress on Evolutionary Computation, CEC'02*, May 12-17, 2002, Honolulu, Hawaii, USA. (2002) 1958–1963
- [96] Kersting, S., Raidl, G., Ljubić, I.: A memetic algorithm for vertex-biconnectivity augmentation. In Cagnoni, S., et al., eds.: *Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim*. Volume 2279 of LNCS., Kinsale, Ireland, Springer-Verlag (2002) 101–110
- [97] Merz, P., Zell, A.: Clustering gene expression profiles with memetic algorithms. In: 7th International Conference on Parallel Problem Solving from Nature - PPSN VII, September 7-11, 2002, Granada, Spain. (2002)
- [98] Buriol, L., Resende, M., Ribeiro, C., Thorup, M.: A memetic algorithm for OSPF routing. In: Sixth INFORMS Telecommunications Conference, March 10-13, 2002 Hilton Deerfield Beach, Boca Raton, Florida. (2002) 187–188
- [99] Cotta, C., Fernández, A.: A hybrid GRASP - evolutionary algorithm approach to golomb ruler search. In Yao, X., et al., eds.: *Parallel Problem Solving From Nature VIII*. Volume 3242 of *Lecture Notes in Computer Science.*, Berlin, Springer-Verlag (2004) 481–490
- [100] Cotta, C., Dotú, I., Fernández, A.J., Hentenryck, P.V.: A memetic approach to golomb rulers. In Runarsson, T., et al., eds.: *Parallel Problem Solving from Nature IX*. Volume 4193 of *Lecture Notes in Computer Science.*, Berlin Heidelberg, Springer-Verlag (2006) 252–261



- [101] Gallardo, J.E., Cotta, C., Fernández, A.J.: A multi-level memetic/exact hybrid algorithm for the still life problem. In Runarsson, T., et al., eds.: *Parallel Problem Solving from Nature IX*. Volume 4193 of *Lecture Notes in Computer Science.*, Berlin Heidelberg, Springer-Verlag (2006) 212–221
- [102] dos Santos Coelho, L., Rudek, M., Junior, O.C.: Fuzzy-memetic approach for prediction of chaotic time series and nonlinear identification. In: *6th On-line World Conference on Soft Computing in Industrial Applications, Organized by World Federation of Soft Computing.* (2001) Co-sponsored by IEEE Systems, Man, and Cybernetics Society.
- [103] Burke, E., Smith, A.: A memetic algorithm to schedule grid maintenance. In: *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation, Vienna: Evolutionary Computation and Fuzzy Logic for Intelligent Control, Knowledge Acquisition and Information Retrieval*, IOS Press (1999) 122–127
- [104] Burke, E., Smith, A.: A multi-stage approach for the thermal generator maintenance scheduling problem. In: *Proceedings of the 1999 Congress on Evolutionary Computation*, Piscataway, NJ, USA, IEEE (1999) 1085–1092
- [105] Cheng, R., Gen, M., Tsujimura, Y.: A tutorial survey of job-shop scheduling problems using genetic algorithms. II. Hybrid genetic search strategies. *Computers & Industrial Engineering* **37** (1999) 51–55
- [106] Sevaux, M., Jouglet, A., Oğuz, C.: Combining constraint programming and memetic algorithm for the hybrid flowshop scheduling problem. In: *ORBEL 19<sup>th</sup> annual conference of the SOGESCI-BVWB*, Louvain-la-Neuve, Belgium (2005)
- [107] Ramat, E., Venturini, G., Lente, C., Slimane, M.: Solving the multiple resource constrained project scheduling problem with a hybrid genetic algorithm. In Bäck, T., ed.: *Proceedings of the Seventh International Conference on Genetic Algorithms*, San Francisco CA, Morgan Kaufmann (1997) 489–496
- [108] Ozdamar, L.: A genetic algorithm approach to a general category project scheduling problem. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* **29** (1999) 44–59
- [109] Watson, J., Rana, S., Whitley, L., Howe, A.: The impact of approximate evaluation on the performance of search algorithms for warehouse scheduling. *Journal of Scheduling* **2** (1999) 79–98
- [110] Dellaert, N., Jeunet, J.: Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm. *International Journal of Production Research* **38** (2000) 1083–1099
- [111] Ming, X., Mak, K.: A hybrid hopfield network-genetic algorithm approach to optimal process plan selection. *International Journal of Production Research* **38** (2000) 1823–1839
- [112] Burke, E.K., Elliman, D.G., Weare, R.F.: A hybrid genetic algorithm for highly constrained timetabling problems. In: *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Francisco, CA (1995) 605–610
- [113] Paechter, B., Cumming, A., Norman, M., Luchian, H.: Extensions to a Memetic timetabling system. In Burke, E., Ross, P., eds.: *The Practice and Theory of Automated Timetabling*. Volume 1153 of *Lecture Notes in Computer Science*. Springer Verlag (1996) 251–265

- [114] de Causmaecker, P., van den Berghe, G., Burke, E.: Using tabu search as a local heuristic in a memetic algorithm for the nurse rostering problem. In: Proceedings of the Thirteenth Conference on Quantitative Methods for Decision Making, Brussels, Belgium (1999) abstract only, poster presentation
- [115] Burke, E.K., De Causmaecker, P., van den Berghe, G.: Novel metaheuristic approaches to nurse rostering problems in belgian hospitals. In Leung, J., ed.: Handbook of Scheduling: Algorithms, Models, and Performance Analysis. Chapman Hall/CRC Press (2004) 44.1–44.18
- [116] Costa, D.: An evolutionary tabu search algorithm and the NHL scheduling problem. *INFOR* **33** (1995) 161–178
- [117] Burke, E.K., Newall, J.: A multi-stage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation* **3** (1999) 63–74
- [118] Gonçalves, J.: A memetic algorithm for the examination timetabling problem. In: Optimization 2001, Aveiro, Portugal, July 23-25, 2001. (2001)
- [119] Ichimura, T., Kuriyama, Y.: Learning of neural networks with parallel hybrid GA using a Royal Road function. In: 1998 IEEE International Joint Conference on Neural Networks. Volume 2., New York, NY, IEEE (1998) 1131–1136
- [120] Topchy, A., Lebedko, O., Miagkikh, V.: Fast learning in multilayered networks by means of hybrid evolutionary and gradient algorithms. In: Proceedings of International Conference on Evolutionary Computation and its Applications. (1996) 390–398
- [121] Aguilar, J., Colmenares, A.: Resolution of pattern recognition problems using a hybrid genetic/random neural network learning algorithm. *Pattern Analysis and Applications* **1** (1998) 52–61
- [122] Krishna, K., Narasimha-Murthy, M.: Genetic  $k$ -means algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* **29** (1999) 433–439
- [123] Mignotte, M., Collet, C., Pérez, P., Bouthemy, P.: Hybrid genetic optimization and statistical model based approach for the classification of shadow shapes in sonar imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000) 129–141
- [124] Ostermark, R.: A neuro-genetic algorithm for heteroskedastic time-series processes: empirical tests on global asset returns. *Soft Computing* **3** (1999) 206–220
- [125] Cotta, C., Troya, J.: Using a hybrid evolutionary-A\* approach for learning reactive behaviors. In Cagnoni, S., et al., eds.: Real-World Applications of Evolutionary Computation. Volume 1803 of Lecture Notes in Computer Science., Edinburgh, Springer-Verlag (2000) 347–356
- [126] Pratihari, D., Deb, K., Ghosh, A.: Fuzzy-genetic algorithms and mobile robot navigation among static obstacles. In: Proceedings of the 1999 Congress on Evolutionary Computation, Washington D.C., IEEE (1999) 327–334
- [127] Ridao, M., Riquelme, J., Camacho, E., Toro, M.: An evolutionary and local search algorithm for planning two manipulators motion. In Del Pobil, A., Mira, J., Ali, M., eds.: Tasks and Methods in Applied Artificial Intelligence. Volume 1416 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg (1998) 105–114

- [128] Chaiyaratana, N., Zalzala, A.: Hybridisation of neural networks and genetic algorithms for time-optimal control. In: Proceedings of the 1999 Congress on Evolutionary Computation, Washington D.C., IEEE (1999) 389–396
- [129] Areibi, S., Moussa, M., Abdullah, H.: A comparison of genetic/memetic algorithms and heuristic searching. In: Proceedings of the 2001 International Conference on Artificial Intelligence ICAI 2001, Las Vegas, Nevada, June 25. (2001)
- [130] Yeh, I.: Hybrid genetic algorithms for optimization of truss structures. *Computer Aided Civil and Infrastructure Engineering* **14** (1999) 199–206
- [131] Pacey, M., Patterson, E., James, M.: A photoelastic technique for characterising fatigue crack closure and the effective stress intensity factor. *Zeszyty Naukowe Politechniki Opolskiej, Seria: Mechanika* **z.67** (2001) *VII Summer School of Fracture Mechanics, Current Research in Fatigue and Fracture*, Pokrzywna (Poland), 18-22 Jun. 2001.
- [132] Wang, L., Yen, J.: Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and kalman filter. *Fuzzy Sets and Systems* **101** (1999) 353–362
- [133] Zelinka, I., Vasek, V., Kolomaznik, K., Dostal, P., Lampinen, J.: Memetic algorithm and global optimization of chemical reactor. In: PC Control 2001, 13th International Conference on Process Control, High Tatras, Slovakia. (2001)
- [134] Knödler, K., Poland, J., Zell, A., Mitterer, A.: Memetic algorithms for combinatorial optimization problems in the calibration of modern combustion engines. In Langdon, W.B., et al., eds.: GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, New York, Morgan Kaufmann Publishers (2002) 687
- [135] Bos, A.: Aircraft conceptual design by genetic/gradient-guided optimization. *Engineering Applications of Artificial Intelligence* **11** (1998) 377–382
- [136] Quagliarella, D., Vicini, A.: Hybrid genetic algorithms as tools for complex optimisation problems. In Blonda, P., Castellano, M., Petrosino, A., eds.: *New Trends in Fuzzy Logic II. Proceedings of the Second Italian Workshop on Fuzzy Logic*, Singapore, World Scientific (1998) 300–307
- [137] Hodgson, R.: Memetic algorithm approach to thin-film optical coating design. In Hart, W., Krasnogor, N., Smith, J., eds.: *Second Workshop on Memetic Algorithms (2nd WOMA)*, San Francisco, California, USA (2001) 152–157
- [138] Srinivasan, D., Cheu, R., Poh, Y., Ng, A.: Development of an intelligent technique for traffic network incident detection. *Engineering Applications of Artificial Intelligence* **13** (2000) 311–322
- [139] Urdaneta, A., Gómez, J., Sorrentino, E., Flores, L., Díaz, R.: A hybrid genetic algorithm for optimal reactive power planning based upon successive linear programming. *IEEE Transactions on Power Systems* **14** (1999) 1292–1298
- [140] Haas, O., Burnham, K., Mills, J.: Optimization of beam orientation in radiotherapy using planar geometry. *Physics in Medicine and Biology* **43** (1998) 2179–2193
- [141] Wehrens, R., Lucasius, C., Buydens, L., Kateman, G.: HIPS, A hybrid self-adapting expert system for nuclear magnetic resonance spectrum interpretation using genetic algorithms. *Analytica Chimica ACTA* **277** (1993) 313–324

- [142] Li, F., Morgan, R., Williams, D.: Economic environmental dispatch made easy with hybrid genetic algorithms. In: Proceedings of the International Conference on Electrical Engineering. Volume 2., Beijing, China, Int. Acad. Publishers (1996) 965–969
- [143] Ostermark, R.: Solving irregular econometric and mathematical optimization problems with a genetic hybrid algorithm. *Computational Economics* **13** (1999) 103–115
- [144] Musil, M., Wilmut, M., Chapman, N.: A hybrid simplex genetic algorithm for estimating geoacoustic parameters using matched-field inversion. *IEEE Journal of Oceanic Engineering* **24** (1999) 358–369
- [145] Reich, C.: Simulation of imprecise ordinary differential equations using evolutionary algorithms. In Carroll, J., Damiani, E., Haddad, H., Oppenheim, D., eds.: ACM Symposium on Applied Computing 2000, ACM Press (2000) 428–432
- [146] Wei, P., Cheng, L.: A hybrid genetic algorithm for function optimization. *Journal of Software* **10** (1999) 819–823
- [147] Wei, X., Kangling, F.: A hybrid genetic algorithm for global solution of nondifferentiable nonlinear function. *Control Theory & Applications* **17** (2000) 180–183
- [148] Cadieux, S., Tanizaki, N., Okamura, T.: Time efficient and robust 3-D brain image centering and realignment using hybrid genetic algorithm. In: Proceedings of the 36th SICE Annual Conference, IEEE (1997) 1279–1284
- [149] Krishna, K., Ramakrishnan, K., Thathachar, M.: Vector quantization using genetic k-means algorithm for image compression. In: 1997 International Conference on Information, Communications and Signal Processing. Volume 3., New York, NY, IEEE (1997) 1585–1587
- [150] Yoneyama, M., Komori, H., Nakamura, S.: Estimation of impulse response of vocal tract using hybrid genetic algorithm—a case of only glottal source. *Journal of the Acoustical Society of Japan* **55** (1999) 821–830
- [151] Merz, P., Zell, A.: Clustering gene expression profiles with memetic algorithms. In Yao, X., et al., eds.: *Parallel Problem Solving From Nature VIII*. Volume 3242 of Lecture Notes in Computer Science., Berlin, Springer-Verlag (2004) 811–820
- [152] Moscato, P., Berretta, R., Mendes, A.: A new memetic algorithm for ordering datasets: Applications in microarray analysis. In Doerner, K., et al., eds.: *Proceedings of the 6th Metaheuristics International Conference*, Vienna, Austria (2005) 695–700
- [153] Knowles, J., Corne, D.: M-PAES: A memetic algorithm for multiobjective optimization. In: *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, La Jolla Marriott Hotel La Jolla, California, USA, IEEE Press (2000) 325–332
- [154] Ponnambalam, S.G., Mohan Reddy, M.: A GA-SA multiobjective hybrid search algorithm for integrating lot sizing and sequencing in flow-line scheduling. *International Journal of Advanced Manufacturing Technology* **21** (2003) 126–137
- [155] Krasnogor, N., Smith, J.E.: Emergence of profitable search strategies based on a simple inheritance mechanism. In Spector, L., et al., eds.: *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*, Morgan Kaufmann (2001) 432–439
- [156] Li, J., Kwan, R.S.K.: A self adjusting algorithm for driver scheduling. *Journal of Heuristics* **11** (2005) 351–367

- [157] Cowling, P.I., Ouelhadj, D., Petrovic, S.: Dynamic scheduling of steel casting and milling using multi-agents. *Production Planning and Control* **15** (2002) 1–11
- [158] Cowling, P.I., Ouelhadj, D., Petrovic, S.: A multi-agent architecture for dynamic scheduling of steel hot rolling. *Journal of Intelligent Manufacturing* **14** (2002) 457–470
- [159] Burke, E.K., Kendall, G., Newall, J., Hart, E., Ross, P., Schulenburg, S.: Hyperheuristics: an emerging direction in modern search technology. In Glover, F., Kochenberger, G., eds.: *Handbook of Metaheuristics*. Kluwer Academic Publishers, Boston MA (2003) 457–474
- [160] Kendall, G., Soubeiga, E., Cowling, P.I.: Choice function and random hyperheuristics. In: *Proceedings of the 4<sup>th</sup> Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*. (2002) 667–671
- [161] Downey, R., Fellows, M.: *Parameterized Complexity*. Springer-Verlag (1998)