

An Introduction to Memetic Algorithms

Pablo Moscato (1), Carlos Cotta (2)

(1) School of Electrical Engineering and Computer Science
University of Newcastle
Callaghan, NSW, 2308, Australia

(2) Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Campus de Teatinos, 29071 Málaga, Spain

e-mail: moscato@cs.newcastle.edu.au, ccottap@cc.uma.es

Memetic algorithms (MAs) are optimization techniques based on the synergistic combination of ideas taken from other metaheuristics, such as populational search (as in evolutionary techniques), and local search (as in gradient-ascent techniques). This work shows the general structure of MAs, and provides some guidelines for their design. Furthermore, an overview of the numerous applications of these techniques is presented. The paper finishes with an sketch of current trends in the development of the area.

Una Introducción a los Algoritmos Meméticos

Pablo Moscato

School of Electrical Engineering and Computer Science,
University of Newcastle, Callaghan, NSW, 2308, Australia
moscato@cs.newcastle.edu.au

Carlos Cotta

Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga, Campus de Teatinos, 29071 - Málaga, España
ccottap@lcc.uma.es

Resumen

Los algoritmos meméticos (MA) son técnicas de optimización que combinan sinérgicamente conceptos tomados de otras metaheurísticas, tales como la búsqueda basada en poblaciones (como en los algoritmos evolutivos), y la mejora local (como en las técnicas de seguimiento del gradiente). En este trabajo se muestra la anatomía general de un MA, y se proporcionan pautas para el diseño de los mismos. Asimismo, se muestra una panorámica general del amplio abanico de aplicaciones que tienen estas técnicas. El trabajo concluye bosquejando algunos de los aspectos de mayor interés actual en el desarrollo del área.

1. Introducción

Los orígenes de los algoritmos meméticos (MA) se remontan a finales de los años ochenta a pesar de que algunos trabajos en décadas anteriores también tienen similares características. En aquella época, el campo de la *computación evolutiva* estaba comenzando a afianzarse sólidamente, una vez que el choque conceptual que había causado la introducción de estas técnicas al mundo de la optimización se iba atenuando. Otro tanto cabía decir de técnicas relacionadas como podían ser el *recocido simulado* (SA) o la *búsqueda tabú* (TS). En general, estas técnicas hacían uso de heurísticas subordinadas para llevar a cabo el proceso de optimización, motivo por el cual se acuñó el término '*metaheurísticas*' para denominarlas (véase el primer artículo de esta monografía).

Fue en este escenario en el que surgió la idea básica que sustenta a los MA: combinar conceptos y estrategias de diferentes metaheurísticas

para intentar aunar las ventajas de las mismas. La denominación 'memético' surge del término inglés '*meme*', acuñado por R. Dawkins como el análogo del *gen* en el contexto de la evolución cultural. La siguiente cita [36] (nuestra traducción) ilustra el concepto:

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

Tras esta definición tan provocadora subyace la idea central de los MA: mejoras individuales de las soluciones en cada uno de los agentes junto con procesos de cooperación y competiciones

de tipo poblacional. En un claro caso de resonancia histórica, los MA¹ tuvieron los mismos difíciles comienzos que los clásicos algoritmos evolutivos (EA) y como éstos, han conseguido vencer las reticencias iniciales a su uso, hasta el punto de haberse convertido hoy en día en un modelo de optimización puntero, con infinidad de aplicaciones prácticas. La Sección 4 dará una breve panorámica de las mismas. Previamente, se verá una definición un tanto más precisa de qué es lo que se entiende por MA. Tal como se mostrará en la Sección 2 a través de una plantilla algorítmica general, un MA puede ser visto como una estrategia de búsqueda en la que un conjunto de *agentes* optimizadores cooperan y compiten [99].

Tras presentar la plantilla algorítmica anteriormente mencionada, se procederá a discutir cómo ésta puede ser instanciada para abordar problemas de optimización concretos. En este sentido, se verá como un MA intenta explotar todo el conocimiento que sobre dichos problemas de optimización se tenga. Esta filosofía de funcionamiento está muy en la línea de lo que L. Davis defendía ya a principios de los 90 [35], y que posteriormente vino a ser confirmado por resultados teóricos bien conocidos, como por ejemplo el Teorema de “No Free Lunch” (NFL) de Wolpert y Macready [136].

Esta explotación del conocimiento sensible que sobre el problema de optimización se tiene puede llevarse a cabo de diferentes formas, e.g., usando heurísticas preexistentes, representaciones *ad hoc*, etc. Estos aspectos serán tratados en la Sección 3. Finalizaremos este trabajo con una breve reseña de algunas de las líneas de investigación más interesantes que en la actualidad están abiertas en el campo de los MA.

2. Anatomía de un MA

Los MA combinan conceptos de diferentes metaheurísticas tal como se adelantó anteriormente. En particular, la relación de estas técnicas con los EA resulta especialmente significativa, al menos desde lo que se podría denominar un punto de vista sintáctico. Esto es fundamen-

talmente debido a la naturaleza *poblacional* de los MA, y por este motivo comenzaremos la descripción de estas técnicas abordando los aspectos heredados de los EA.

Un MA mantiene en todo momento una *población* de diversas soluciones al problema considerado. Llamaremos *agente* a cada una de las mismas. Esta denominación es una extensión del término *individuo* tan comúnmente empleado en el contexto de los EA, y permite capturar elementos distintivos de los MA que se mostrarán más adelante (por ejemplo, un agente puede contener más de una solución al problema considerado [6]). Estos agentes se interrelacionan entre sí en un marco de competición y de cooperación, de manera muy semejante a lo que ocurre en la Naturaleza entre los individuos de una misma especie (los seres humanos, sin ir más lejos). Cuando consideramos la población de agentes en su conjunto, esta interacción puede ser estructurada en una sucesión de grandes pasos temporales denominados *generaciones*. La Figura 1 muestra el esquema de una de estas generaciones.

```

PROC Paso-Generacional
  (↑ pob : Agente[ ], ↓ ops : Operador[ ])
Variables
  criadores, nueva_pob : Agente[ ];
Inicio
  criadores ← Seleccionar(pob);
  nueva_pob ← Reproducir(criadores, ops);
  pob ← Actualizar (pob, nueva_pob)
Fin

```

Figura 1: Esquema de una generación. En esta figura y en todas las posteriores, los parámetros de entrada y entrada/salida se marcan respectivamente con ↓ y ↓↑.

Tal como se muestra, cada generación consiste en la actualización de la población de agentes, usando para tal fin una nueva población obtenida mediante la recombinación de las características de algunos agentes seleccionados. Precisamente, este componente de selección es, junto con el paso final de actualización, el responsable de forzar la competición entre agentes. Más concretamente, la selección se encarga de elegir una muestra de los mejores agentes contenidos en la población actual. Esto se realiza mediante el empleo de una función guía F_g encargada de cuantificar cuán bueno es cada uno de los agentes en la resolución del problema

¹Los términos EA híbrido, y EA Lamarckiano se usan también frecuentemente para referirse a estas técnicas

```

FUNC Reproducir
  ( $\downarrow pob : Agente[ ], \downarrow ops : Operador[ ]$ )  $\rightarrow$ 
   $\rightarrow Agente[ ]$ 
Variables
   $buf : Agente[ ] [ ]$ ;
   $j : \mathbb{N}$ ;
Inicio
   $buf[0] \leftarrow pob$ ;
  PARA  $j \leftarrow 1 : |ops|$  HACER
     $buf[j] \leftarrow \text{Aplicar}(ops[j], buf[j - 1])$ ;
  FINPARA;
  DEVOLVER  $buf[|ops|]$ 
Fin

```

Figura 2: Generación de la nueva población.

abordado. Por su parte, el reemplazo o actualización incide en el aspecto competitivo, encargándose de la importante tarea de limitar el tamaño de la población, esto es, eliminar algunos agentes para permitir la entrada de otros nuevos y así enfocar la tarea de búsqueda. En este proceso también puede emplearse la información proporcionada por la función guía para seleccionar los agentes que se eliminarán.

Tanto la selección como el reemplazo son procesos puramente competitivos en los que únicamente varía la distribución de agentes existentes, esto es, no se crean nuevos agentes. Esto es responsabilidad de la fase de reproducción. Dicha reproducción tiene lugar mediante la aplicación de cierto número de *operadores* reproductivos tal como se muestra en la Figura 2. Como puede apreciarse, es posible emplear un número variado de operadores. No obstante, lo más típico es emplear únicamente dos operadores: recombinación y mutación. El primero es el responsable de llevar a cabo los procesos de cooperación entre agentes (usualmente dos, aunque es posible considerar grupos más grandes [41]). Dicha cooperación tiene lugar mediante la construcción de nuevos agentes empleando información extraída del grupo de agentes recombinados, y quizás alguna información externa (si bien esto último suele ser en ocasiones algo que se intenta evitar, e.g., ver [77]). Precisamente, la inclusión de información externa no contenida en ninguno de los agentes involucrados es responsabilidad del operador de mutación. Básicamente, este tipo de operador generará un nuevo agente mediante la modificación parcial de un agente existente.

```

PROC Optimizador-Local
  ( $\downarrow actual : Agente, \downarrow op : Operador$ )
Variables
   $nuevo : Agente$ 
Inicio
  REPETIR
     $nuevo \leftarrow \text{Aplicar}(op, actual)$ ;
    SI ( $F_g(nuevo) \prec_{\mathcal{F}} F_g(actual)$ ) ENTONCES
       $actual \leftarrow nuevo$ ;
    FINSI
  HASTA QUE Terminacion-Local();
  DEVOLVER  $actual$ ;
Fin

```

Figura 3: Esquema de un optimizador local

En relación con los operadores de mutación, es posible definir un *meta-operador* basado en la aplicación iterativa de un operador de mutación arbitrario sobre un agente. El empleo de estos metaoperadores es uno de los rasgos más distintivos de los MA. Concretamente, dichos metaoperadores iteran la aplicación del operador de mutación, conservando los cambios que llevan a una mejora en la bondad del agente, motivo por el cual son denominados *optimizadores locales*. La Figura 3 ilustra el proceso. Como puede verse, la iteración en la aplicación del operador de mutación y la subsiguiente conservación de los cambios favorables se realiza hasta que se alcanza un cierto criterio de terminación (un número de iteraciones fijo, un cierto número de iteraciones sin mejora, haber alcanzado una mejora suficiente, etc.).

Estos optimizadores locales pueden considerarse como un operador más, y como tales pueden emplearse en diferentes fases de la reproducción. Por ejemplo, pueden usarse tras la aplicación de otros operadores “simples” de recombinación y mutación, aplicarse sólo a un subconjunto de los agentes o únicamente ser aplicado al final del ciclo reproductivo. En cualquier caso, es la existencia de estos optimizadores individuales la que justifica el empleo de la denominación ‘agente’ en la jerga de los MA: El algoritmo puede caracterizarse como una colección de agentes que realizan exploraciones autónomas del espacio de búsqueda, cooperando ocasionalmente a través de la recombinación, y compitiendo continuamente por los recursos computacionales a través de la selección y el reemplazo ².

²Como puede apreciarse, el empleo de optimización

Tras haber descrito el paso generacional básico, podemos ahora considerar una visión global de un MA basada en la iteración de este procedimiento, tal como se muestra en la Figura 4.

```

FUNC MA ( $\downarrow$  tamañoPob :  $\mathbb{N}$ ,
            $\downarrow$  ops : Operador[ ])  $\rightarrow$  Agente
Variables
  pob : Agente[ ];
Inicio
  pob  $\leftarrow$  Iniciar-Poblacion(tamañoPob);
  REPETIR
    pob  $\leftarrow$  Paso-Generacional (pob, ops)
    SI Convergencia(pob) ENTONCES
      pob  $\leftarrow$  Reiniciar-Poblacion(pob);
    FINSI
  HASTA QUE Terminacion-MA()
  DEVOLVER i-esimo-Mejor(pob, 1);
Fin

```

Figura 4: El esqueleto genérico de un MA

La generación de la población inicial puede acometerse de diferentes formas. Por ejemplo, pueden crearse $|pob|$ agentes al azar, o emplear las soluciones proporcionadas por heurísticas existentes [76][126]. Una posibilidad más sofisticada es el empleo de optimizadores locales para tal fin, tal como se ilustra en la Fig. 5:

```

FUNC Iniciar-Poblacion
  ( $\downarrow$   $\mu$  :  $\mathbb{N}$ )  $\rightarrow$  Agente[ ]
Variables
  pob : Agente[ ];
  agente : Agente;
  j :  $\mathbb{N}$ ;
Inicio
  PARA j  $\leftarrow$  1: $\mu$  HACER
    agente  $\leftarrow$  Agente-Aleatorio();
    pob[j]  $\leftarrow$  Optimizador-Local (agente);
  FINPARA
  DEVOLVER pob
Fin

```

Figura 5: Generación guiada de la población inicial.

La función para la reiniciación de la población es otro de los componentes fundamentales del MA, ya que de él dependerá que se haga un uso apropiado de los recursos computacionales del sistema, o por el contrario éstos se malgasten

local es un elemento diferenciador de los MA, pero ni es el único, ni es imprescindible encontrarlo en la forma descrita. Debe pues evitarse caer en la falsa igualdad MA = EA+búsqueda local.

en la explotación de una población de agentes que haya alcanzado un estado degenerado, e.g., con una gran similitud entre todos los agentes de la población. Esto es lo que se conoce como *convergencia* del MA, y es algo que puede ser cuantificado empleando por ejemplo medidas clásicas de Teoría de la Información como la entropía de Shannon [34]. Una vez se ha detectado la convergencia del MA (cuando la entropía ha caído por debajo de un cierto valor límite por ejemplo), la población de agentes se reinicia, conservando una porción de la misma, y generando nuevos agentes para completarla (Figura 6).

```

FUNC Reiniciar-Poblacion
  ( $\downarrow$  pob : Agente[ ])  $\rightarrow$  Agente[ ]
Variables
  nueva_pob : Agente[ ];
  j, cons :  $\mathbb{N}$ ;
Inicio
  cons  $\leftarrow$  |pob|. %CONSERVAR;
  PARA j  $\leftarrow$  1:cons HACER
    newpop[j]  $\leftarrow$  i-esimo-Mejor(pob, j);
  FINPARA
  PARA j  $\leftarrow$  (cons + 1) : |pob| HACER
    nueva_pob[j]  $\leftarrow$  Agente-Aleatorio();
    nueva_pob[j]  $\leftarrow$ 
       $\leftarrow$  Optimizador-Local (nueva_pob[j]);
  FINPARA;
  DEVOLVER nueva_pob
Fin

```

Figura 6: Reiniciación de la población en un MA.

La función anterior completa la definición funcional de un MA. A modo de resumen, la Tabla 1 muestra una comparativa de los conceptos ms relevantes de los MA, y cómo estos difieren del modelo más clásico de los GA.

3. Técnicas para el Diseño de MA

Cuando se aborda el diseño de un MA efectivo para un cierto problema, hay que partir de la base de que no existe procedimiento sistemático para tal fin, ya que de lo contrario se entraría en conflicto con resultados teóricos bien conocidos (e.g., el Teorema NFL). Ello implica que únicamente pueden considerarse heurísticas de diseño, que probablemente resultarán en un MA

Cuadro 1: Comparativa de algunos de los aspectos más distintivos de los GA y los MA.

GA	
<i>codificación</i>	esquemas, cadenas lineales, alfabetos predefinidos
<i>individuo</i>	solución al problema
<i>cruce</i>	intercambio no-guiado de información
<i>mutación</i>	introducción aleatoria de nueva información
MA	
<i>representación</i>	formas, no-linealidad, cercanía al problema
<i>agente</i>	solución/ones al problema + mecanismo mejora local
<i>recombinación</i>	intercambio guiado de información
<i>mutación</i>	introducción sensible de nueva información
<i>mejora local</i>	aprendizaje lamarckiano

efectivo, pero que obviamente no lo pueden garantizar.

El primer elemento que resulta imprescindible determinar es la representación de las soluciones al problema. Por ejemplo, supóngase que se está intentando resolver una instancia del problema del viajante de comercio (TSP). Informalmente, las soluciones son rutas cerradas que visitan n ciudades sólo una vez. La cuestión en este punto es *qué características de estas soluciones manipularán los operadores reproductivos*. Para empezar, es posible expresar las rutas como una permutación de las ciudades, y definir operadores que manipulen los valores existentes en posiciones específicas de la permutación. Esta sería la *representación basada en posiciones* de las permutaciones [27][47]. Por otro lado, los operadores podrían manipular información relativa a la adyacencia entre elementos de las permutaciones. Dado que el TSP se define a partir de una matriz de distancia entre pares de ciudades, es intuitivo observar que este último enfoque sería más conveniente. De hecho, puede comprobarse que esto es así en la práctica, y que operadores basados en adyacencia [77] funcionan mejor sobre este problema que otros basados en posiciones tales como PMX [48] o CX [101].

El ejemplo anterior del TSP ilustra la necesidad de capturar de alguna manera cuantitativa

la relación que existe entre una representación de un problema y su bondad. A tal efecto, se han definido diferentes criterios como los que a continuación se mencionan. Algunos de los mismos son generalizaciones de conceptos propios de los GA, mientras que otros sí pueden ser más específicos del modelo MA:

- *Minimización de la epistasis*: se habla de epistasis cuando los elementos básicos de información a partir de los cuales se construyen las soluciones (y que son manipulados por los operadores reproductivos) interactúan de manera no aditiva sobre la función guía [33]. La existencia de una interacción de este tipo impide que se pueda descomponer la función objetivo en términos optimizables de manera independiente³. Obviamente, cuanto menor sea esta interacción no aditiva, mayor será también la relevancia que en términos absolutos tengan los elementos de información manipulados por los operadores, y menos propensa será la función guía a extraviar la búsqueda.
- *Minimización de la varianza de bondad* [113]: La varianza en bondad de un cierto elemento de información es la varianza en los valores que devuelve la función guía, medida sobre un conjunto representativo de soluciones con dicho elemento de información. De manera similar a lo que ocurría en el criterio anterior, cuanto menor sea esta varianza más significativos serán los elementos de información, y habrá menos “ruido” en la función guía.
- *Maximización de la correlación de bondad*: asumiendo un cierto operador reproductivo, se mide la correlación existente entre la adecuación de los agentes *progenitores* y los agentes *descendientes*. Si la correlación es alta, los agentes buenos tendrán una descendencia buena por lo general. Dado que la fase de selección se encarga de elegir a los agentes de mayor adecuación, la búsqueda se irá desplazando gradualmente hacia las zonas más prometedoras del espacio de búsqueda.

³Este impedimento ha de entenderse en el sentido de que una optimización separada de los elementos básicos de información conduciría a un óptimo local de la función objetivo.

En cualquier caso, lo que resulta claro de la discusión anterior es que en último extremo la selección de una representación es algo que depende de los operadores empleados. Es pues la selección de operadores el problema de diseño que debe abordarse. En este sentido, existen dos vertientes: la selección de un operador de entre un conjunto de operadores preexistentes, o la definición de nuevos operadores. En el primer caso, la línea de actuación podría ser la que sigue [27]:

1. Sea $\Omega = \{\omega_1, \omega_2, \dots, \omega_k\}$ el conjunto de operadores disponibles. En primer lugar se identifica la representación manipulada por cada uno de ellos.
2. Usar cualquiera de los criterios anteriormente mencionados para evaluar la bondad de cada representación.
3. Seleccionar el operador ω_i que manipule la representación de mayor bondad.

Esto se denomina *análisis inverso de operadores*, ya que en cierto sentido se realiza una cierta ingeniería inversa para determinar el operador/representación más ventajoso. La alternativa es el *análisis directo*, en el que se diseñan nuevos operadores como sigue [27]:

1. Identificar varias representaciones para el problema considerado.
2. Usar cualquiera de los criterios anteriormente mencionados para evaluar la bondad de cada representación.
3. Crear nuevos operadores $\Omega' = \{\omega'_1, \omega'_2, \dots, \omega'_m\}$ a través de la manipulación de los elementos de información más relevantes.

El último paso del análisis directo puede realizarse empleando para tal fin algunas plantillas genéricas (independientes de los elementos de información manipulados) diseñadas a tal fin, e.g., “*random respectful recombination*” (R^3), “*Random Assorting Recombination*” (RAR), “*Random Transmitting Recombination*” (RTR), etc. [112].

Tanto en el caso de selección de un operador “clásico” como de creación de un opera-

dor a partir de las plantillas genéricas mencionadas anteriormente, se estarán empleando típicamente operadores “ciegos”, que manipulan información relevante pero lo hacen sin usar información de la instancia del problema que se pretende resolver. Obviamente resultaría muy interesante ser capaces de introducir conocimiento adicional para poder guiar esta manipulación. A los operadores reproductivos que usan este conocimiento del problema se les denomina *heurísticos* o *híbridos* (hasta en cierto caso se los tildó de *meméticos*), pudiendo considerarse dos ámbitos principales en los que utilizar dicho conocimiento: la selección de la información de los progenitores que se usará en la descendencia, y la selección de la información externa que se empleará.

Con respecto a lo primero, la evidencia experimental sugiere que la propiedad de *respeto* (transmisión de características comunes a todos los progenitores) es beneficiosa (e.g., véase [28] [77]). Tras esta transmisión inicial de información, la descendencia puede completarse de diferentes maneras. Por ejemplo, Radcliffe y Surry [113] proponen el empleo de optimizadores locales, o técnicas de enumeración implícita. En ambos casos, el método elegido se aplicaría para completar las soluciones empleando cuanta información externa fuera necesaria.

En el caso particular de utilizar técnicas de enumeración implícita (por ejemplo, ramificación y acotación), el enfoque anterior puede ser computacionalmente costoso. Una alternativa es el empleo de dicha técnica de enumeración implícita, restringida al conjunto de información contenida en los agentes progenitores. Este enfoque se denomina *recombinación dinásticamente óptima* [31] (DOR), y tiene la propiedad de ser monótono creciente en la bondad de las soluciones generadas, i.e., cualquier descendiente es al menos tan bueno como el mejor de sus progenitores.

El conocimiento del problema puede incluirse también en los operadores mediante el empleo de heurísticas constructivas. Por ejemplo, EAX (*Edge Assembly Crossover*) [97] es un operador especializado para el TSP en el que se emplea una heurística voraz para la construcción de la descendencia.

Las ideas expuestas en los párrafos anteriores son aplicables a los operadores de mutación,

aunque con ciertas salvedades, ya que estos están forzados a introducir nueva información. En general, nótese que la solución parcial a la cual se ha hecho mención anteriormente puede obtenerse mediante la eliminación de algunos elementos de información de una solución. Ello quiere decir que mediante la posterior aplicación de un procedimiento de completado de los ya mencionados se obtendría una solución mutada.

Como cierre de esta sección, es importante reseñar una vez más el carácter heurístico de los principios de diseño mencionados. Hay aún mucho por hacer en esta vertiente metodológica de los MA (e.g., véase [64]). En este sentido, nótese que la filosofía de los MA es muy flexible, y desde siempre se orientó a la incorporación de ideas tomadas de otras metaheurísticas. Por ello, del estudio de las diferentes técnicas de optimización discutidas en esta monografía pueden obtenerse nuevas ideas y herramientas conducentes a nuevos y más potentes algoritmos meméticos.

4. Aplicaciones de los MA

Existen cientos de aplicaciones de los MA en el ámbito de la optimización combinatoria. 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. Curiosamente, es frecuente que estos MA se hallen *disfrazados* u *ocultos* bajo otras denominaciones (EA/GA híbridos, EA/GA lamarckianos, etc.), tal como se mencionó en la Sección 1. No obstante, esta falta de consistencia terminológica es cada vez menos frecuente.

Hecha esta apreciación, ofrecemos aquí una breve lista de algunas de las aplicaciones que dan cuenta de la popularidad que ha alcanzado esta metodología. Destacamos los siguientes: *problemas de particionado en grafos* [83] [138], *partición de números* [7], *conjunto Independiente de cardinalidad máxima* [1] [54] [122], *empaquetado* [117], *coloreado de grafos* [26] [39] [43] [24], *recubrimiento de conjuntos* [4], *planificación de tareas en una máquina con tiempos de "set-up" y fechas de entrega* [44] [87], *planificación de tareas en varias máquinas* [22]

[78] [88], *problemas de asignación generalizados* [23], *problemas de mochila multidimensional* [5] [29] [51], *programación entera no-lineal* [127] *asignación cuadrática* [18] [81] [82], *particionado de conjuntos* [72], y muy especialmente el *problema del viajante de comercio* [50] [56] [82] [92] [114].

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* [105], *optimización en "paisajes NK"* [80], *diseño de trayectorias óptimas para naves espaciales* [32], *asignación de frecuencias* [59], *construcción de árboles de expansión mínimos con restricciones de grado* [115], *problemas de emplazamiento* [57] [71] [123], *optimización de rutas* [120], *problemas de transporte* [46] [100], *asignación de tareas* [53], *isomorfismos en grafos* [129], *problemas de biconexión de vértices* [60], *agrupamiento* [85], *telecomunicaciones* [10] [111] [121], *identificación de sistemas no-lineales* [40], *optimización multiobjetivo* [62], *programación de tareas de mantenimiento* [13] [14] [15], *"open shop scheduling"* [21] [42] [74], *"flowshop scheduling"* [19] [94] [95], *planificación de proyectos* [98] [116] [106], *planificación de almacén* [132], *planificación de producción* [38] [89] [8], *confección de horarios* [11] [75] [108], *planificación de turnos* [37], *planificación de juegos deportivos* [25] y *planificación de exámenes* [12] [49].

Los MA, también han sido citados en la literatura de aprendizaje en máquinas y robótica como algoritmos genéticos híbridos. Destacamos algunas aplicaciones como por ejemplo: *entrenamiento de redes neuronales* [58] [90] [128], *reconocimiento de características* [2], *clasificación de características* [69] [86], *análisis de series temporales* [103], *aprendizaje de comportamientos reactivos en agentes móviles* [30], *planificación de trayectorias* [102] [109] [119], *control óptimo* [20], etc.

En las áreas de la Electrónica y la Ingeniería

podemos destacar: *proyectos de VLSI* [3], *optimización de estructuras* [137] y *mecánica de fracturas* [107], *modelado de sistemas* [131], *control de reactores químicos* [140], *calibración de motores* [61], *problemas de diseño óptimo en Aeronáutica* [9] [110], *diseño de sistemas ópticos* [55], *control de tráfico* [125], y *planificación en problemas de potencia* [130] entre otros.

Otras aplicaciones de estas técnicas pueden encontrarse en: *Medicina* [52] [133], *Economía* [73] [104], *Oceanografía* [96], *Matemáticas* [118] [134] [135], *Procesamiento de imágenes y de voz* [16] [70] [139], y un sinnúmero de ellas en Biología Molecular.

5. Conclusiones

Existen dos fuerzas importantes que favorecen la aplicación de MA en varias áreas. Por un lado, la creciente disponibilidad de sistemas de cómputación concurrente, generalmente basados en clusters, permite a los investigadores la posibilidad de paralelizar con cierta facilidad los programas. Los MA se adaptan muy bien a este tipo de paralelismo, a lo que hay que añadir la creciente relevancia de lenguajes como JAVA, que facilitan aún más esta tarea. Por otro lado, ya existe una mejor comprensión, al menos heurística, sobre como crear MA eficientes. A ello se suman ciertos avances recientes en la teoría de la complejidad computacional de operadores de recombinación.

Existe también un creciente interés en el desarrollo sistemático de los MA mediante la incorporación de otras técnicas. Estas pueden ser de dos tipos: exactas (e.g., algoritmos de ramificación y poda, algoritmos de tratabilidad a parámetro fijo, etc.), o metaheurística (como la incorporación de periodos de Búsqueda Tabú para la diversificación, etc.). Un ejemplo clásico puede encontrarse en [7]: un método muy básico de Búsqueda Tabú potencia un MA y permite obtener muy buenas soluciones en un problema donde las técnicas de Recocido Simulado tuvieron grandes dificultades.

Existen abiertas algunas líneas muy interesantes en Investigación, como por ejemplo en la co-evolución. Citando [91]:

“Es posible que una generación futura de MA trabaje en dos niveles y dos escalas temporales. En una escala de corto alcance, un conjunto de agentes trabajaría en el espacio de búsqueda asociado al problema mientras que en una escala de más largo plazo, los agentes adaptarían las heurísticas a ellos asociadas. Nuestro trabajo con D. Holstein que será presentado en este libro puede ser clasificado como un primer paso en esta dirección prometedora. Sin embargo, es razonable pensar que en breve se implementarán de manera práctica nuevos y mas complejos esquemas, involucrando soluciones, agentes y también representaciones.”

En el trabajo con D. Holstein, fue empleado un MA que también utilizaba *Búsqueda Local Guiada*. Algunos artículos mas recientes están en esta dirección [17][63][67]. Krasnogor introdujo la terminología de “*multimeme*” para inequívocamente identificar MA que también adaptan la definición de vecindario utilizado en las búsquedas locales [68], y con colegas está aplicando el método en el difícil problema de la predicción de las estructuras secundarias de las proteínas [65]. Otro estudio de estos temas puede encontrarse en [124].

Otras áreas de interés en el desarrollo de MA están en el uso de estructuras de la población [45][93], búsquedas locales auto-adaptables [66] y nuevos operadores de recombinación siguiendo las metodologías empleadas en [56][79][84][90].

Referencias

- [1] C.C. Aggarwal, J.B. Orlin, and R.P. Tai. Optimized crossover for the independent set problem. *Operations Research*, 45(2):226–234, 1997.
- [2] J. Aguilar and A. Colmenares. Resolution of pattern recognition problems using a hybrid genetic/random neural network learning algorithm. *Pattern Analysis and Applications*, 1(1):52–61, 1998.
- [3] S. Areibi, M. Moussa, and H. Abdullah. A comparison of genetic/memetic algo-

- rithms and heuristic searching. In *Proceedings of the 2001 International Conference on Artificial Intelligence ICAI 2001, Las Vegas, Nevada, June 25, 2001*.
- [4] J. Beasley and P.C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2):393–404, 1996.
- [5] J. Beasley and P.C. Chu. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4:63–86, 1998.
- [6] R. Berretta, C. Cotta, and P. Moscato. Enhancing the performance of memetic algorithms by using a matching-based recombination algorithm: Results on the number partitioning problem. In M. Resende and J. Pinho de Sousa, editors, *Metaheuristics: Computer-Decision Making*, pages 65–90. Kluwer Academic Publishers, Boston MA, 2003.
- [7] R. Berretta and P. Moscato. The number partitioning problem: An open challenge for evolutionary computation? In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 261–278. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999.
- [8] R. Berretta and L. F. Rodrigues. A memetic algorithm for multi-stage capacitated lot-sizing problems. *International Journal of Production Economics*. En prensa.
- [9] A.H.W. Bos. Aircraft conceptual design by genetic/gradient-guided optimization. *Engineering Applications of Artificial Intelligence*, 11(3):377–382, 1998.
- [10] L. Buriol, M.G.C. Resende, C.C. Ribeiro, and M. Thorup. A memetic algorithm for OSPF routing. In *Sixth INFORMS Telecommunications Conference, March 10-13, 2002 Hilton Deerfield Beach, Boca Raton, Florida*, pages 187–188, 2002.
- [11] E. K. Burke, D. G. Elliman, and R. F. Weare. A hybrid genetic algorithm for highly constrained timetabling problems. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 605–610. Morgan Kaufmann, San Francisco, CA, 1995.
- [12] E.K. Burke, J.P. Newall, and R.F. Weare. A memetic algorithm for university exam timetabling. In E.K. Burke and P. Ross, editors, *The Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 241–250. Springer-Verlag, 1996.
- [13] E.K. Burke and A.J. Smith. A memetic algorithm for the maintenance scheduling problem. In *Proceedings of the ICONIP/ANZIIS/ANNES '97 Conference*, pages 469–472, Dunedin, New Zealand, 1997. Springer-Verlag.
- [14] E.K. Burke and A.J. Smith. 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*, pages 122–127. IOS Press, 1999.
- [15] E.K. Burke and A.J. Smith. A multi-stage approach for the thermal generator maintenance scheduling problem. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1085–1092, Piscataway, NJ, USA, 1999. IEEE.
- [16] S. Cadieux, N. Tanizaki, and T. Okamura. Time efficient and robust 3-D brain image centering and realignment using hybrid genetic algorithm. In *Proceedings of the 36th SICE Annual Conference*, pages 1279–1284. IEEE, 1997.
- [17] R. Carr, W. Hart, N. Krasnogor, J. Hirst, E. Burke, and J. Smith. Alignment of protein structures with a memetic evolutionary algorithm. In W. B. Langdon et al., editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1027–1034, New York, 9-13 July 2002. Morgan Kaufmann Publishers.
- [18] J. Carrizo, F.G. Tinetti, and P. Moscato. A computational ecology for the quadratic assignment problem. In *Proceedings of the 21st Meeting on Informatics and Operations Research*, Buenos Aires, Argentina, 1992. SADIO.

- [19] S. Cavalieri and P. Gaiardelli. Hybrid genetic algorithms for a multiple-objective scheduling problem. *Journal of Intelligent Manufacturing*, 9(4):361–367, 1998.
- [20] N. Chaiyaratana and A.M.S. Zalzala. Hybridisation of neural networks and genetic algorithms for time-optimal control. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 389–396, Washington D.C., 1999. IEEE.
- [21] R. Cheng, M. Gen, and Y. Tsujimura. A tutorial survey of job-shop scheduling problems using genetic algorithms. II. Hybrid genetic search strategies. *Computers & Industrial Engineering*, 37(1-2):51–55, 1999.
- [22] R.W. Cheng and M. Gen. Parallel machine scheduling problems using memetic algorithms. *Computers & Industrial Engineering*, 33(3-4):761–764, 1997.
- [23] P.C. Chu and J. Beasley. A genetic algorithm for the generalised assignment problem. *Computers & Operations Research*, 24:17–23, 1997.
- [24] P.E. Coll, G.A. Durán, and P. Moscato. On worst-case and comparative analysis as design principles for efficient recombination operators: A graph coloring case study. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 279–294. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999.
- [25] D. Costa. An evolutionary tabu search algorithm and the NHL scheduling problem. *INFOR*, 33(3):161–178, 1995.
- [26] D. Costa, N. Dubuis, and A. Hertz. Embedding of a sequential procedure within an evolutionary algorithm for coloring problems in graphs. *Journal of Heuristics*, 1(1):105–128, 1995.
- [27] C. Cotta. A study of hybridisation techniques and their application to the design of evolutionary algorithms. *AI Communications*, 11(3-4):223–224, 1998.
- [28] C. Cotta and J. Muruzábal. Towards a more efficient evolutionary induction of bayesian networks. In J.J. Merelo et al., editors, *Parallel Problem Solving From Nature VII*, volume 2439 of *Lecture Notes in Computer Science*. Springer-Verlag, Paris, 2002.
- [29] C. Cotta and J.M. Troya. A hybrid genetic algorithm for the 0-1 multiple knapsack problem. In G.D. Smith, N.C. Steele, and R.F. Albrecht, editors, *Artificial Neural Nets and Genetic Algorithms 3*, pages 251–255, Wien New York, 1998. Springer-Verlag.
- [30] C. Cotta and J.M. Troya. Using a hybrid evolutionary-A* approach for learning reactive behaviors. In S. Cagnoni et al., editors, *Real-World Applications of Evolutionary Computation*, volume 1803 of *Lecture Notes in Computer Science*, pages 347–356, Edinburgh, 15-16 April 2000. Springer-Verlag.
- [31] C. Cotta and J.M. Troya. Embedding branch and bound within evolutionary algorithms. *Applied Intelligence*, 18:137–153, 2003.
- [32] T. Crain, R. Bishop, W. Fowler, and K. Rock. Optimal interplanetary trajectory design via hybrid genetic algorithm/recursive quadratic program search. In *Ninth AAS/AIAA Space Flight Mechanics Meeting*, pages 99–133, Breckenridge CO, 1999.
- [33] Y. Davidor. Epistasis variance: A viewpoint on GA-hardness. In G.J.E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 23–35. Morgan Kaufmann, 1991.
- [34] Y. Davidor and O. Ben-Kiki. The interplay among the genetic algorithm operators: Information theory tools used in a holistic way. In R. Männer and B. Manderick, editors, *Parallel Problem Solving From Nature II*, pages 75–84, Amsterdam, 1992. Elsevier Science Publishers B.V.
- [35] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold Computer Library, New York, 1991.
- [36] R. Dawkins. *The Selfish Gene*. Clarendon Press, Oxford, 1976.
- [37] P. de Causmaecker, G. van den Berghe, and E.K. Burke. 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*, pages abstract only, poster presentation, Brussels, Belgium, 1999.
- [38] N. Dellaert and J. Jeunet. Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm. *International Journal of Production Research*, 38(5):1083–1099, 2000.
- [39] R. Dorne and J.K. Hao. A new genetic local search algorithm for graph coloring. In A.E. Eiben, Th. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving From Nature V*, volume 1498 of *Lecture Notes in Computer Science*, pages 745–754, Berlin, 1998. Springer-Verlag.
- [40] L. dos Santos Coelho, M. Rudek, and O. Canciglieri Junior. 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.
- [41] A.E. Eiben, P.-E. Raue, and Zs. Ruttkay. Genetic algorithms with multi-parent recombination. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving From Nature III*, volume 866 of *Lecture Notes in Computer Science*, pages 78–87. Springer-Verlag, 1994.
- [42] J. Fang and Y. Xi. A rolling horizon job shop rescheduling strategy in the dynamic environment. *International Journal of Advanced Manufacturing Technology*, 13(3):227–232, 1997.
- [43] C. Fleurent and J.A. Ferland. Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research*, 63:437–461, 1997.
- [44] P.M. França, A.S. Mendes, and P. Moscato. 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*, pages 1708–1710, Atlanta, GA, USA, 1999. Decision Sciences Institute.
- [45] P.M. França, A.S. Mendes, and P. Moscato. A memetic algorithm for the total tardiness single machine scheduling problem. *European Journal of Operational Research*, 132(1):224–242, 2001.
- [46] M. Gen, K. Ida, and L. Yinzheng. Bicriteria transportation problem by hybrid genetic algorithm. *Computers & Industrial Engineering*, 35(1-2):363–366, 1998.
- [47] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- [48] D.E. Goldberg and R. Lingle Jr. Alleles, loci and the traveling salesman problem. In J.J. Grefenstette, editor, *Proceedings of an International Conference on Genetic Algorithms*, Hillsdale NJ, 1985. Lawrence Erlbaum Associates.
- [49] J.F. Gonçalves. A memetic algorithm for the examination timetabling problem. In *Optimization 2001, Aveiro, Portugal, July 23-25, 2001*, 2001.
- [50] M. Gorges-Schleuter. *Genetic Algorithms and Population Structures - A Massively Parallel Algorithm*. PhD thesis, University of Dortmund, Germany, 1991.
- [51] J. Gottlieb. Permutation-based evolutionary algorithms for multidimensional knapsack problems. In J. Carroll, E. Damiani, H. Haddad, and D. Openheim, editors, *ACM Symposium on Applied Computing 2000*, pages 408–414. ACM Press, 2000.
- [52] O.C.L. Haas, K.J. Burnham, and J.A. Mills. Optimization of beam orientation in radiotherapy using planar geometry. *Physics in Medicine and Biology*, 43(8):2179–2193, 1998.
- [53] A.B. Hadj-Alouane, J.C. Bean, and K.G. Murty. A hybrid genetic/optimization algorithm for a task allocation problem. *Journal of Scheduling*, 2(4), 1999.
- [54] M. Hifi. A genetic algorithm-based heuristic for solving the weighted maximum independent set and some equivalent

- lent problems. *Journal of the Operational Research Society*, 48(6):612–622, 1997.
- [55] R.J.W. Hodgson. Memetic algorithm approach to thin-film optical coating design. In W.E. Hart, N. Krasnogor, and J. Smith, editors, *Second Workshop on Memetic Algorithms (2nd WOMA)*, pages 152–157, San Francisco, California, USA, 7 July 2001.
- [56] D. Holstein and P. Moscato. Memetic algorithms using guided local search: A case study. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 235–244. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999.
- [57] E. Hopper and B. Turton. A genetic algorithm for a 2d industrial packing problem. *Computers & Industrial Engineering*, 37(1-2):375–378, 1999.
- [58] T. Ichimura and Y. Kuriyama. Learning of neural networks with parallel hybrid GA using a Royal Road function. In *1998 IEEE International Joint Conference on Neural Networks*, volume 2, pages 1131–1136, New York, NY, 1998. IEEE.
- [59] I.E. Kassotakis, M.E. Markaki, and A.V. Vasilakos. A hybrid genetic approach for channel reuse in multiple access telecommunication networks. *IEEE Journal on Selected Areas in Communications*, 18(2):234–243, 2000.
- [60] S. Kersting, G.R. Raidl, and I. Ljubić. A memetic algorithm for vertex-biconnectivity augmentation. In S. Cagnoni et al., editors, *Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim*, volume 2279 of *LNCS*, pages 101–110, Kinsale, Ireland, 3-4 April 2002. Springer-Verlag.
- [61] K. Knödler, J. Poland, A. Zell, and A. Mitterer. Memetic algorithms for combinatorial optimization problems in the calibration of modern combustion engines. In W. B. Langdon et al., editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, page 687, New York, 9-13 July 2002. Morgan Kaufmann Publishers.
- [62] J. Knowles and D. Corne. M-PAES: A memetic algorithm for multiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 325–332, La Jolla Marriott Hotel La Jolla, California, USA, 6-9 July 2000. IEEE Press.
- [63] N. Krasnogor. Coevolution of genes and memes in memetic algorithms. In Unam-May O’Reilly, editor, *Graduate Student Workshop*, page 371, Orlando, Florida, USA, 13 July 1999.
- [64] N. Krasnogor. *Studies on the Theory and Design Space of Memetic Algorithms*. Ph.D. Thesis, Faculty of Engineering, Computer Science and Mathematics. University of the West of England. Bristol, United Kingdom, 2002.
- [65] N. Krasnogor, B. Blackburne, J.D. Hirst, and E.K. Burke. Multimeme algorithms for protein structure prediction. In *7th International Conference on Parallel Problem Solving from Nature - PPSN VII, September 7-11, 2002, Granada, Spain*, 2002.
- [66] N. Krasnogor and J. Smith. A memetic algorithm with self-adaptive local search: TSP as a case study. In Darrell Whitley et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 987–994, Las Vegas, Nevada, USA, 10-12 July 2000. Morgan Kaufmann.
- [67] N. Krasnogor and J. Smith. Emergence of profitable search strategies based on a simple inheritance mechanism. In L. Spector et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 432–439, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.
- [68] N. Krasnogor and J. Smith. Multimeme algorithms for the structure prediction and structure comparison of proteins. In Alwyn M. Barry, editor, *GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference*, pages 42–44, New York, 8 July 2002. AAAI.

- [69] K. Krishna and M. Narasimha-Murty. Genetic k -means algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, 1999.
- [70] K. Krishna, K.R. Ramakrishnan, and M.A.L. Thathachar. Vector quantization using genetic k -means algorithm for image compression. In *1997 International Conference on Information, Communications and Signal Processing*, volume 3, pages 1585–1587, New York, NY, 1997. IEEE.
- [71] R.M. Krzanowski and J. Raper. Hybrid genetic algorithm for transmitter location in wireless networks. *Computers, Environment and Urban Systems*, 23(5):359–382, 1999.
- [72] D. Levine. A parallel genetic algorithm for the set partitioning problem. In I.H. Osman and J.P. Kelly, editors, *Meta-Heuristics: Theory & Applications*, pages 23–35. Kluwer Academic Publishers, Boston, MA, USA, 1996.
- [73] F. Li, R. Morgan, and D. Williams. Economic environmental dispatch made easy with hybrid genetic algorithms. In *Proceedings of the International Conference on Electrical Engineering*, volume 2, pages 965–969, Beijing, China, 1996. Int. Acad. Publishers.
- [74] C.F. Liaw. A hybrid genetic algorithm for the open shop scheduling problem. *European Journal of Operational Research*, 124(1):28–42, 2000.
- [75] S. E. Ling. Integrating genetic algorithms with a prolog assignment program as a hybrid solution for a polytechnic timetable problem. In *Parallel Problem Solving from Nature II*, pages 321–329. Elsevier Science Publisher B. V., 1992.
- [76] S.J. Louis, X. Yin, and Z.Y. Yuan. Multiple vehicle routing with time windows using genetic algorithms. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1804–1808, Washington D.C., 1999. IEEE Neural Network Council - Evolutionary Programming Society - Institution of Electrical Engineers.
- [77] K. Mathias and D. Whitley. Genetic operators, the Fitness Landscape and the Traveling Salesman Problem. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature - Proceedings of 2nd Workshop, PPSN II*, pages 219–228. Elsevier Science Publishers, 1992.
- [78] A.S. Mendes, F.M. Muller, P.M. França, and P. Moscato. 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, pages 1–6, Campinas, SP, Brazil, 1999. Technological Center for Informatics Foundation.
- [79] P. Merz. A comparison of memetic recombination operators for the traveling salesman problem. In W. B. Langdon et al., editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 472–479, New York, 9-13 July 2002. Morgan Kaufmann Publishers.
- [80] P. Merz and B. Freisleben. On the Effectiveness of Evolutionary Search in High-Dimensional NK -Landscapes. In D. Fogel, editor, *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, pages 741–745, Piscataway, NJ, USA, 1998. IEEE Press.
- [81] P. Merz and B. Freisleben. A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem. In Pete Angeline, editor, *1999 Congress on Evolutionary Computation (CEC'99)*, pages 2063–2070, Piscataway, NJ, USA, 1999. IEEE Press.
- [82] P. Merz and B. Freisleben. Fitness landscapes and memetic algorithm design. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 245–260. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999.
- [83] P. Merz and B. Freisleben. Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning.

- Evolutionary Computation*, 8(1):61–91, 2000.
- [84] P. Merz and B. Freisleben. Memetic algorithms for the traveling salesman problem. *Complex Systems*, 132(42):297–345, 2001.
- [85] P. Merz and A. Zell. 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.
- [86] M. Mignotte, C. Collet, P. Pérez, and P. Boutheymy. 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(2):129–141, 2000.
- [87] D.M. Miller, H.C. Chen, J. Matson, and Q. Liu. A hybrid genetic algorithm for the single machine scheduling problem. *Journal of Heuristics*, 5(4):437–454, 1999.
- [88] L. Min and W. Cheng. Identical parallel machine scheduling problem for minimizing the makespan using genetic algorithm combined with simulated annealing. *Chinese Journal of Electronics*, 7(4):317–321, 1998.
- [89] X.G. Ming and K.L. Mak. A hybrid hopfield network-genetic algorithm approach to optimal process plan selection. *International Journal of Production Research*, 38(8):1823–1839, 2000.
- [90] P. Moscato. An Introduction to Population Approaches for Optimization and Hierarchical Objective Functions: The Role of Tabu Search. *Annals of Operations Research*, 41(1-4):85–121, 1993.
- [91] P. Moscato. Memetic algorithms: A short introduction. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 219–234. McGraw-Hill, Maidenhead, Berkshire, England, UK, 1999.
- [92] P. Moscato and M. G. Norman. A Memetic Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems. In M. Valero, E. Onate, M. Jane, J. L. Larriba, and B. Suarez, editors, *Parallel Computing and Transputer Applications*, pages 177–186, Amsterdam, 1992. IOS Press.
- [93] P. Moscato and F. Tinetti. Blending heuristics with a population-based approach: A memetic algorithm for the traveling salesman problem. Report 92-12, Universidad Nacional de La Plata, C.C. 75, 1900 La Plata, Argentina, 1992.
- [94] T. Murata and H. Ishibuchi. Performance evaluation of genetic algorithms for flowshop scheduling problems. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, volume 2, pages 812–817, New York, NY, 1994. IEEE.
- [95] T. Murata, H. Ishibuchi, and H. Tanaka. Genetic algorithms for flowshop scheduling problems. *Computers & Industrial Engineering*, 30(4):1061–1071, 1996.
- [96] M. Musil, M.J. Wilmot, and N.R. Chapman. A hybrid simplex genetic algorithm for estimating geoacoustic parameters using matched-field inversion. *IEEE Journal of Oceanic Engineering*, 24(3):358–369, 1999.
- [97] Y. Nagata and Sh. Kobayashi. Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. In Th. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, EE.UU.*, pages 450–457, San Mateo, CA, 1997. Morgan Kaufmann.
- [98] A.L. Nordstrom and S. Tufekci. A genetic algorithm for the talent scheduling problem. *Computers & Operations-Research*, 21(8):927–940, 1994.
- [99] M.G. Norman and P. Moscato. A competitive and cooperative approach to complex combinatorial search. Technical Report Caltech Concurrent Computation Program, Report. 790, California Institute of Technology, Pasadena, California, USA, 1989. expanded version published

- at the Proceedings of the 20th Informatics and Operations Research Meeting, Buenos Aires (20th JAIIO), Aug. 1991, pp. 3.15–3.29.
- [100] A.G.N. Novaes, J.E.S. De-Cursi, and O.D. Gracioli. A continuous approach to the design of physical distribution systems. *Computers & Operations Research*, 27(9):877–893, 2000.
- [101] I.M. Oliver, D.J. Smith, and J.R.C. Holland. A study of permutation crossover operators on the traveling salesperson problem. In J.J. Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms and their Applications*, pages 224–230, Hillsdale NJ, 1987. Lawrence Erlbaum Associates.
- [102] P. Osmera. Hybrid and distributed genetic algorithms for motion control. In V. Chundy and E. Kurekova, editors, *Proceedings of the Fourth International Symposium on Measurement and Control in Robotics*, pages 297–300, 1995.
- [103] R. Ostermark. A neuro-genetic algorithm for heteroskedastic time-series processes: empirical tests on global asset returns. *Soft Computing*, 3(4):206–220, 1999.
- [104] R. Ostermark. Solving irregular economic and mathematical optimization problems with a genetic hybrid algorithm. *Computational Economics*, 13(2):103–115, 1999.
- [105] E. Ozcan and C.K. Mohan. Steady state memetic algorithm for partial shape matching. In V.W. Porto, N. Saravanan, and D. Waagen, editors, *Evolutionary Programming VII*, volume 1447 of *Lecture Notes in Computer Science*, pages 527–536. Springer, Berlin, 1998.
- [106] L. Ozdamar. A genetic algorithm approach to a general category project scheduling problem. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 29(1):44–59, 1999.
- [107] M.N. Pacey, E.A. Patterson, and M.N. James. A photoelastic technique for characterising fatigue crack closure and the effective stress intensity factor. *Zeszyty Naukowe Politechniki Opolskiej, Seria: Mechanika*, z.67(Nr kol. 269/2001), 2001. *VII Summer School of Fracture Mechanics, Current Research in Fatigue and Fracture*, Pokrzywna (Poland), 18-22 Jun. 2001.
- [108] B. Paechter, A. Cumming, M.G. Norman, and H. Luchian. Extensions to a Memetic timetabling system. In E.K. Burke and P. Ross, editors, *The Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 251–265. Springer Verlag, 1996.
- [109] D.K. Pratihar, K. Deb, and A. Ghosh. Fuzzy-genetic algorithms and mobile robot navigation among static obstacles. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 327–334, Washington D.C., 1999. IEEE.
- [110] D. Quagliarella and A. Vicini. Hybrid genetic algorithms as tools for complex optimisation problems. In P. Blonda, M. Castellano, and A. Petrosino, editors, *New Trends in Fuzzy Logic II. Proceedings of the Second Italian Workshop on Fuzzy Logic*, pages 300–307, Singapore, 1998. World Scientific.
- [111] A. Quintero and S. Pierre. A multi-population memetic algorithm to optimize the assignment of cells to switches in cellular mobile networks, 2001. submitted for publication.
- [112] N.J. Radcliffe. The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 10:339–384, 1994.
- [113] N.J. Radcliffe and P.D. Surry. Fitness Variance of Formae and Performance Prediction. In L.D. Whitley and M.D. Vose, editors, *Proceedings of the 3rd Workshop on Foundations of Genetic Algorithms*, pages 51–72, San Francisco, 1994. Morgan Kaufmann.
- [114] N.J. Radcliffe and P.D. Surry. Formal Memetic Algorithms. In T. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, volume 865 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, Berlin, 1994.

- [115] G.R. Raidl and B.A. Julstron. A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem. In J. Carroll, E. Damiani, H. Haddad, and D. Oppenheim, editors, *ACM Symposium on Applied Computing 2000*, pages 440–445. ACM Press, 2000.
- [116] E. Ramat, G. Venturini, C. Lente, and M. Slimane. Solving the multiple resource constrained project scheduling problem with a hybrid genetic algorithm. In Th. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 489–496, San Francisco CA, 1997. Morgan Kaufmann.
- [117] C. Reeves. Hybrid genetic algorithms for bin-packing and related problems. *Annals of Operations Research*, 63:371–396, 1996.
- [118] C. Reich. Simulation of imprecise ordinary differential equations using evolutionary algorithms. In J. Carroll, E. Damiani, H. Haddad, and D. Oppenheim, editors, *ACM Symposium on Applied Computing 2000*, pages 428–432. ACM Press, 2000.
- [119] M.A. Ridao, J. Riquelme, E.F. Camacho, and M. Toro. An evolutionary and local search algorithm for planning two manipulators motion. In A.P. Del Pobil, J. Mira, and M. Ali, editors, *Tasks and Methods in Applied Artificial Intelligence*, volume 1416 of *Lecture Notes in Computer Science*, pages 105–114. Springer-Verlag, Berlin Heidelberg, 1998.
- [120] A.M. Rodrigues and J. Soeiro Ferreira. Solving the rural postman problem by memetic algorithms. In Jorge Pinho de Sousa, editor, *Proceedings of the 4th Metaheuristic International Conference (MIC'2001), Porto, Portugal, July 16-20, 2001*, pages 679–684, 2001.
- [121] S. Runggeratigul. A memetic algorithm to communication network design taking into consideration an existing network. In Jorge Pinho de Sousa, editor, *Proceedings of the 4th Metaheuristic International Conference (MIC'2001), Porto, Portugal, July 16-20, 2001*, pages 91–96, 2001.
- [122] A. Sakamoto, X.Z. Liu, and T. Shimamoto. A genetic approach for maximum independent set problems. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E80A(3):551–556, 1997.
- [123] V. Schnecke and O. Vornberger. Hybrid genetic algorithms for constrained placement problems. *IEEE Transactions on Evolutionary Computation*, 1(4):266–277, 1997.
- [124] Jim Smith. Co-evolving memetic algorithms: Initial investigations. In *7th International Conference on Parallel Problem Solving from Nature - PPSN VII, September 7-11, 2002, Granada, Spain*, 2002.
- [125] D. Srinivasan, R.L. Cheu, Y.P. Poh, and A.K.C. Ng. Development of an intelligent technique for traffic network incident detection. *Engineering Applications of Artificial Intelligence*, 13(3):311–322, 2000.
- [126] P.D. Surry and N.J. Radcliffe. Inoculation to initialise evolutionary search. In T.C. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, number 1143 in *Lecture Notes in Computer Science*, pages 269–285. Springer-Verlag, 1996.
- [127] T. Taguchi, T. Yokota, and M. Gen. Reliability optimal design problem with interval coefficients using hybrid genetic algorithms. *Computers & Industrial Engineering*, 35(1–2):373–376, 1998.
- [128] A.P. Topchy, O.A. Lebedko, and V.V. Miagkikh. Fast learning in multilayered networks by means of hybrid evolutionary and gradient algorithms. In *Proceedings of International Conference on Evolutionary Computation and its Applications*, pages 390–398, June 1996.
- [129] R. Torres-Velazquez and V. Estivill-Castro. A memetic algorithm instantiated with selection sort consistently finds global optima for the error-correcting graph isomorphism. In Xin Yao, editor, *Proceedings of the IEEE 2002 Congress on Evolutionary Computation, CEC'02, May 12-17, 2002, Honolulu, Hawaii, USA*, pages 1958–1963, 2002.

- [130] A.J. Urdaneta, J.F. Gómez, E. Sorrentino, L. Flores, and R. Díaz. A hybrid genetic algorithm for optimal reactive power planning based upon successive linear programming. *IEEE Transactions on Power Systems*, 14(4):1292–1298, 1999.
- [131] L. Wang and J. Yen. Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and kalman filter. *Fuzzy Sets and Systems*, 101(3):353–362, 1999.
- [132] J.P. Watson, S. Rana, L.D. Whitley, and A.E. Howe. The impact of approximate evaluation on the performance of search algorithms for warehouse scheduling. *Journal of Scheduling*, 2(2):79–98, 1999.
- [133] R. Wehrens, C. Lucasius, L. Buydens, and G. Kateman. HIPS, A hybrid self-adapting expert system for nuclear magnetic resonance spectrum interpretation using genetic algorithms. *Analytica Chimica ACTA*, 277(2):313–324, May 1993.
- [134] P. Wei and L.X. Cheng. A hybrid genetic algorithm for function optimization. *Journal of Software*, 10(8):819–823, 1999.
- [135] X. Wei and F. Kangling. A hybrid genetic algorithm for global solution of nondifferentiable nonlinear function. *Control Theory & Applications*, 17(2):180–183, 2000.
- [136] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.
- [137] I.C. Yeh. Hybrid genetic algorithms for optimization of truss structures. *Computer Aided Civil and Infrastructure Engineering*, 14(3):199–206, 1999.
- [138] W.-C. Yeh. A memetic algorithm for the min k -cut problem. *Control and Intelligent Systems*, 28(2):47–55, 2000.
- [139] M. Yoneyama, H. Komori, and S. Nakamura. 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(12):821–830, 1999.
- [140] I. Zelinka, V. Vasek, K. Kolomaznik, P. Dostal, and J. Lampinen. Memetic algorithm and global optimization of chemical reactor. In *PC Control 2001, 13th International Conference on Process Control, High Tatras, Slovakia*, 2001.