

Aplicación de las Técnicas de Procesamiento de Alta Performance a la Implementación de Algoritmos Genéticos

Reseña del estado del arte y
evolución de las clasificaciones de
Algoritmos Genéticos Paralelos

TR 02-06

Sergio Nesmachnow
Instituto de Computación
Facultad de Ingeniería
Universidad de la República
Uruguay

Julio 2002

Resumen

El objetivo de este trabajo consiste en brindar una descripción de los mecanismos de aplicación de las técnicas de procesamiento de alta performance propuestas por la comunidad científica para mejorar el desempeño y la calidad de búsqueda de los algoritmos genéticos. El trabajo conforma una reseña del estado del arte orientada a exponer la evolución en la clasificación de los enfoques de paralelismo aplicados a los algoritmos genéticos en los últimos veinticinco años.

Aplicación de las Técnicas de Procesamiento de Alta Performance a la Implementación de Algoritmos Genéticos

"Evolution is cleverer than you are."
Francis Crick [Den84]

1 - Introducción

Este trabajo describe los mecanismos de aplicación de las técnicas de procesamiento de alta performance (procesamiento paralelo y distribuido) propuestas por los investigadores en el área de la computación evolutiva, con el objetivo de mejorar el desempeño y la calidad de búsqueda de los algoritmos genéticos (AG).

El trabajo constituye en sí mismo una reseña del estado del arte en el área de los algoritmos genéticos paralelos, elaborada desde el punto de vista de la evolución en la clasificación de los enfoques de paralelismo aplicados.

El documento se organiza del modo que se detalla a continuación. Una descripción del formalismo de los algoritmos genéticos y los motivos para la aplicación de técnicas de procesamiento paralelo se ofrece en el capítulo 2. En este mismo capítulo se explica la evolución histórica identificada en las clasificaciones de Algoritmos Genéticos Paralelos (AGP). Los períodos de dicha evolución se analizan en los capítulos 3 al 5, incluyendo un breve resumen de las ideas y categorizaciones principales, en base al análisis realizado sobre los trabajos más importantes de cada etapa.

Como conclusión, el capítulo 6 ofrece un panorama del estado actual de desarrollo del área, y la identificación de los diferentes modelos de AGP sobre la propuesta de taxonomía unificada elaborada en la etapa más reciente.

Este trabajo fue elaborado en el marco del proyecto "Algoritmos Genéticos Paralelos y su Aplicación al Diseño de Redes de Comunicaciones Confiables"¹ desarrollado en el Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay.

¹ Proyecto financiado con fondos de CSIC (Comisión Sectorial de Investigación Científica)

2 – Algoritmos genéticos y técnicas de programación de alta performance

Algoritmos genéticos

Los algoritmos evolutivos basan su funcionamiento en un mecanismo análogo a los procesos evolutivos que ocurren en la naturaleza, con el objetivo de resolver problemas de búsqueda y optimización.

Para el caso de los Algoritmos Genéticos (AG), durante el proceso evolutivo se mantiene una población de soluciones que evolucionan de acuerdo a operaciones de selección, cruzamiento, reemplazo y mutación, siguiendo la idea de la supervivencia de los individuos más aptos. El grado de adaptación de un individuo se evalúa de acuerdo al problema a resolver, mediante la definición de una *función de fitness*.

Una descripción detallada del mecanismo operativo de los AG puede encontrarse en el texto de Goldberg [Gol89a].

Motivos para la aplicación de las técnicas de procesamiento de alta performance

A pesar de que el mecanismo de los AG se basa en un proceso de evolución que es paralelo por naturaleza, el modelo clásico de AG propuesto originalmente por Holland [Hol76] es secuencial, motivado por la falta de disponibilidad de arquitecturas paralelas al momento de su definición. Como consecuencia de esta restricción, algunos aspectos del funcionamiento del AG clásico resultan difíciles o imposibles de paralelizar, como es el caso de la selección panmíctica utilizada para imitar el concepto de presión selectiva. Otros mecanismos no existentes en el AG serial surgen al considerar los modelos paralelos, como el operador de migración entre poblaciones semi-independientes.

Las técnicas de procesamiento paralelo y distribuido se aplican al modelo clásico de los AG con el objetivo de obtener mejoras tanto desde el punto de vista de la performance del mecanismo evolutivo como desde el punto de vista algorítmico, para mejorar la calidad de la búsqueda genética.

Analizando desde el punto de vista de la eficiencia, la paralelización permite afrontar la lentitud de la convergencia de los AG cuando la dimensión del problema hace necesario utilizar poblaciones numerosas, o en los casos que requieren múltiples evaluaciones de funciones de fitness costosas.

Conjuntamente a la posibilidad de obtener mejores tiempos de ejecución, mediante la aplicación de técnicas de procesamiento paralelo es posible obtener mejor precisión y eficiencia numérica en los resultados, tal como se describe en los trabajos de Gordon & Whitley [GoW93], Hart et al. [Har96], entre otros.

Este hecho indica que la propia estructura algorítmica de los Algoritmos Genéticos Paralelos (AGP) constituye un tópico interesante, independientemente de su implementación sobre una arquitectura de alta performance específica.

Desde el punto de vista algorítmico, los AGP pueden explotar la naturaleza paralela intrínseca del mecanismo evolutivo, trabajando simultáneamente sobre varias poblaciones independientes o semi-independientes para resolver el mismo problema. Eventuales intercambios de soluciones (*migraciones*) permiten introducir la diversidad y variación necesaria para evitar los problemas de convergencia prematura en valores sub-óptimos. Complementariamente, los AGP pueden aprovechar las características de paralelismo implícitas en el propio problema de optimización, analizando concurrentemente diferentes secciones del espacio de búsqueda del problema.

Cómo aplicar las técnicas de programación de alta performance

Análogamente a todo problema susceptible de resolución mediante las técnicas de procesamiento de alta performance, dos enfoques principales surgen intuitivamente para la paralelización del mecanismo de los AG. Estos enfoques se corresponden con las técnicas usuales del procesamiento paralelo, las técnicas de descomposición funcional y descomposición de dominio, y han estado presentes en los trabajos de investigación desde los primeros intentos por paralelizar los AG.

El modelo que surge de distribuir funcionalmente el algoritmo, asignando a diferentes procesadores distintas etapas del mecanismo evolutivo, se conoce como modelo maestro-esclavo. La funcionalidad candidata a paralelizar en un AG es la evaluación de la función de fitness, la cual frecuentemente involucra un tiempo de ejecución mayor que el de los sencillos operadores evolutivos.

Por contrapartida, un enfoque orientado a la distribución de datos da lugar a un modelo de población distribuida, organizada en subpoblaciones semi-independientes.

Estos dos modelos, reconocidos desde las primeras etapas de la investigación de los AGP, han evolucionado y se han diversificado. De modo particular, el enfoque de población distribuida ha dado lugar a varios modelos de paralelismo aplicado a los AG.

Usualmente al aplicar una técnica de procesamiento paralelo a un algoritmo serial, se obtiene un algoritmo paralelo, el cual provisto con la misma entrada que el algoritmo serial original, produce exactamente la misma salida que este.

Excepto para el modelo maestro-esclavo, en el cual sólo se distribuye la evaluación de la función de fitness, los modelos de AGP violan la regla de producir la misma salida con la misma entrada que el algoritmo serial. Dado que el modelo de interacción panmíctica se reemplaza por un mecanismo de poblaciones semi-independientes, el comportamiento del algoritmo distribuido es diferente. En una población panmíctica cada individuo tiene la posibilidad de interactuar –competir o cruzarse– con cualquier otro, lo que no ocurre para el modelo de poblaciones distribuidas en donde las interacciones están limitadas a individuos de la misma población o vecindad.

Esta característica peculiar de los AGP de poblaciones distribuidas proporciona nuevos matices de interés sobre el problema paralelizado, dependiendo de los diferentes criterios utilizados al dividir el problema.

Criterios de clasificación

Un análisis de las ideas de aplicación de las técnicas de procesamiento paralelo y distribuido a los AG, permite establecer diferentes criterios para clasificar los diseños propuestos de AGP. De acuerdo a la literatura especializada, pueden identificarse los criterios que se ofrecen a continuación:

- El modo de evaluación de la función de fitness, que puede ser concentrada o distribuirse en diferentes procesos.
- El número de poblaciones utilizadas por el AGP. Las alternativas consisten en utilizar una única población (modelo panmíctico) o utilizar poblaciones múltiples.
- El tamaño de las subpoblaciones y su organización, para aquellos modelos de poblaciones múltiples.
- El mecanismo de intercambio de individuos entre poblaciones múltiples.
- El modo de aplicación del mecanismo de cruzamiento, que puede ser centralizado o distribuido.
- El conjunto de individuos considerado para la aplicación del mecanismo de selección, puede ser un conjunto local o global.
- El mecanismo de sincronización entre los elementos de procesamiento.

El principal criterio de clasificación utilizado por los investigadores en el área considera el modo de organizar la población de individuos solución del problema durante el proceso evolutivo.

Etapas en el desarrollo - La "evolución" en las clasificaciones

En los últimos años, significativos avances en el ámbito del procesamiento paralelo y de la programación evolutiva han consolidado a los AGP como una técnica de optimización robusta y eficiente para la resolución de problemas NP-difíciles.

Desde sus orígenes a mediados de la década de 1970, una gran cantidad de propuestas de implementación y modelos teóricos han sido presentados por los investigadores en el área de la computación evolutiva y el procesamiento paralelo.

Analizando su contenido, es posible distinguir tres etapas en la investigación, relevantes para comprender los conceptos teóricos, de diseño e implementación de los AGP.

Conjuntamente, con el paso de los años las propuestas de clasificación se han extendido para considerar la totalidad de los modelos de aplicación de paralelismo a los AG.

Un esquema cronológico de las etapas identificadas se ofrece en la figura 1.

La primera etapa corresponde a los orígenes del modelo, en la cual fueron realizados análisis teóricos, propuestas de diseño y las primeras implementaciones de AGP. Cronológicamente, esta etapa se extiende desde los primeros estudios sobre AG realizados a mediados de la década de 1970 hasta el fin de la década de 1980.

La segunda etapa se caracteriza por una prolífica gama de implementaciones de AGP aplicados a problemas particulares, la mayoría implementados sobre plataformas específicas de hardware de alta performance. Esta etapa se extendió a lo largo de la primera mitad de la década de 1990 y estuvo claramente influenciada por la difusión de las técnicas de procesamiento paralelo, en particular sobre multicomputadores de memoria compartida y por la popularización de las técnicas de procesamiento distribuido sobre redes de computadoras.

La etapa que se extiende desde la segunda mitad de la década de 1990 hasta la actualidad se distingue por la preocupación por parte de los investigadores por la formalización de los conceptos teóricos y el modelado matemático del mecanismo de los AGP como herramienta de optimización. Complementariamente, se ha avanzado en la comprensión de los modos de aplicación de las técnicas de procesamiento paralelo, hasta proponerse taxonomías completas, que constituyen un intento de unificar la clasificación de los AGP.

En los siguientes capítulos se resumen las características de cada una de estas etapas y se analizan los modelos propuestos, comentando especialmente los trabajos considerados representativos del avance en cada período.

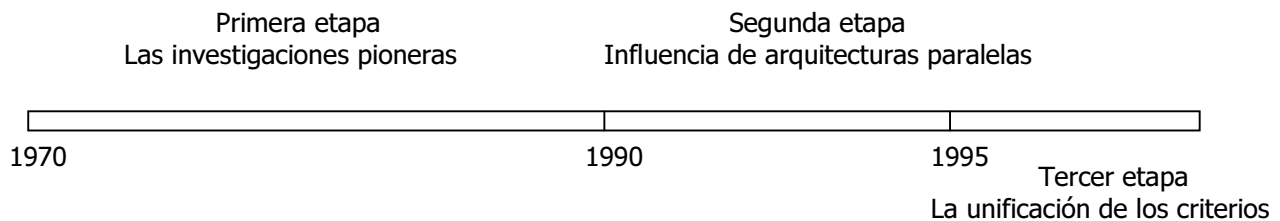


Figura 1 - Etapas en el desarrollo de AGP

3 - Las investigaciones pioneras

La naturaleza paralela del mecanismo de los sistemas adaptativos y de los algoritmos evolutivos fue reconocida en los trabajos pioneros en el área por parte de Holland [Hol63], [Hol65] y Bossert [Bos67] en la primera mitad de la década de 1960. De acuerdo a la cita de Goldberg [Gol89a], Holland inclusive discutió el funcionamiento de mecanismos reproductivos sobre un modelo primitivo de máquina paralela, la *computadora de circuitos iterativos*; mientras que Bossert introdujo un modelo de poblaciones múltiples, con competición por supervivencia entre poblaciones, y eventos ocasionales para explorar la totalidad del espacio de soluciones, eliminando la peor población a intervalos aleatorios o modificando la función de fitness con el tiempo.

En sus trabajos, ni Holland ni Bossert llegaron a proponer implementaciones paralelas ni distribuidas de los AG. Sus análisis se concentraron en los modelos seriales, cuyo enfoque permitía el estudio de sus propiedades matemáticas. Estos estudios pioneros facilitaron la comprensión de las características del mecanismo evolutivo y su aplicabilidad como herramienta de optimización.

Desde la citada idea de Holland, deben recorrerse 15 años para encontrar una nueva referencia a la conjunción de la programación evolutiva, algoritmos genéticos en particular, y las técnicas de procesamiento paralelo. En 1975, De Jong [DeJ75] analizó teóricamente el modelo de paralelismo maestro-esclavo sincrónico, y propuso su variante con salto generacional, en los que se define un valor de salto $G \in [1/n, 1]$ que determina la fracción de la población que el AG reemplaza en cada generación.

En 1976, Bethke [Bet76] realizó el estudio teórico de un caso particular de AGP sobre una máquina paralela SIMD, analizando la eficiencia de uso de la capacidad de procesamiento. Bethke concluye que la eficiencia se acerca al máximo posible en los casos en los cuales las evaluaciones de la función de fitness son mucho más costosas que las operaciones de evolución, un caso típico para muchas aplicaciones. Sin embargo, no se incluyó en el análisis la influencia del overhead de comunicación.

Si bien Bethke fue el primero en describir abstractamente la codificación de un AGP según el modelo maestro-esclavo, tampoco él llegó a proponer específicamente una implementación de un algoritmo paralelo, ni a realizar su simulación mediante algoritmos seriales, limitándose al análisis teórico mencionado.

La primera referencia a una taxonomía de los AGP corresponde a Grefenstette en su trabajo de 1981 [Gre81]. En la clasificación elaborada por este autor se reconocen tres categorías, el modelo maestro-esclavo basado en la distribución de la función de fitness (del cual describe una versión sincrónica y una asincrónica), un modelo para máquinas paralelas de memoria compartida y un modelo distribuido de poblaciones múltiples.

A grandes rasgos, estas categorías primordiales han evolucionado a lo largo del tiempo, refinándose para aplicarse a diseños específicos y/o generalizándose al plantear taxonomías de amplio espectro.

En su trabajo de 1985, Grosso [Gro85] observó que la evolución de la función de fitness era más rápida en una población dividida en subpoblaciones o *demes* que en una única población distribuida y propuso técnicas de migración "retrasadas", en las cuales los demes evolucionan aisladamente hasta llegar a un punto cercano a la convergencia y luego comienzan las migraciones con alta frecuencia. Esta propuesta minimiza el costo debido a las comunicaciones, permitiendo obtener resultados de calidad similar a los obtenidos con las técnicas de migración estándar.

En 1987 se presenta la primera implementación de un AGP, designado con el genérico nombre de *Parallel Genetic Algorithm* (PGA). Este algoritmo se basa en un modelo de subpoblaciones homogéneas cooperativas ("*islas*") con topología de migración dinámica, y fue propuesto por Pettey et al. [Pet87]. El estudio preliminar de esta implementación fue empírico, siendo analizado teóricamente dos años más tarde por dos de sus creadores, Pettey & Leuze [Pet89]. En este último trabajo, los autores amplían los análisis de DeJong [DeJ75] sobre los AG secuenciales con el fin de considerar los efectos de las estrategias de selección, cruzamiento y mutación sobre la eficiencia del PGA. Análisis posteriores del funcionamiento del algoritmo y su éxito probabilístico fueron realizados por Mühlenbein [Müh91] quien lo aplica a la resolución del TSP.

Los AGP de *grano fino* fueron introducidos como una implementación de AGP sobre computadores masivamente paralelos por Robertson en 1987 [Rob87], quien los implementó sobre un equipo Connection Machine 1, paralelizando todas las fases del algoritmo (selección de progenitores, selección de población reemplazada, cruzamiento y mutación). Su implementación seleccionaba y reemplazaba una pequeña fracción de la población en cada generación, y reportó muy buenos resultados, con un tiempo de ejecución independiente del tamaño de la población.

Simultáneamente, los AGP de grano fino fueron estudiados por Manderick & Spiessen [Man89], Mühlenbein [Müh88] y [Müh89] y Gorges-Schleuter [Gor89] proponiendo fijar los individuos en una grilla planar y restringiendo la selección y el cruzamiento a pequeñas vecindades en la grilla.

El modelo de Mühlenbein y Gorges-Schleuter, de nombre ASPARAGOS, fue implementado en su primer versión sobre una red de transputers, utilizando para la población una estructura de escalera cíclica. Luego evolucionó incluyendo diferentes estructuras y mecanismos de pareo [Gor92] hasta constituirse en una herramienta efectiva de optimización [Gor97].

Variantes al modelo genérico de islas propuesto por Pettey et al. fueron introducidas por Cohoon et al. [Coh89] y Tanese [Tan89a]. Cohoon investigó las similitudes entre la teoría de los equilibrios puntuados y el mecanismo evolutivos de los AGP multi-población.

Por su parte, Tanese inició el estudio de los efectos de la migración en la eficiencia y calidad de resultados de los AGP, mostrando que tasas medianas de migración son necesarias para hallar soluciones de la misma calidad que un AG serial. En términos de eficiencia del AGP, concluyó que pueden obtenerse speedups casi lineales cuando la migración es infrecuente y de tasa baja.

Complementariamente, Tanese estudió cómo los parámetros de migración influyen en la pérdida de diversidad de alelos [Tan89b]. Para prevenirla, propuso fijar diferentes parámetros y/o mecanismos de selección y cruzamiento para cada deme, causando que algunos converjan más lentamente que otros, en uno de las primeras referencias de un AGP heterogéneo encontradas en la literatura.

En los experimentos comparativos realizados por Tanese está implícita la idea de que es posible realizar una comparación justa entre pequeñas poblaciones y una población global.

En su texto de 1989 [Gol89a], Goldberg sugiere un diseño "basado en objetos" para los AGP, y describe dos modelos: el modelo de *comunidad* (un híbrido con multipoblaciones y migración en el cual cada población utiliza un modelo distribuido de evaluación de la función de fitness) y el modelo de *polinización* (un modelo distribuido, con estructura asignada a la población que determina la difusión de material genético).

Sus modelos resultaron bastante complejos y no fueron implementados en la práctica. Complementariamente, Goldberg estudió el problema de determinar el tamaño adecuado para las subpoblaciones paralelas en su trabajo de 1989 [Gol89b].

En los trabajos de Grefenstette de 1989 [Gre89] y 1991 [Gre91], se analizan propiedades invariantes para los diferentes modelos de AG propuestos hasta el momento. Se define la categoría de AG *admisibles*, como aquellos que cumplen ciertos requerimientos mínimos en la manera de mapear la función objetivo en una medida de fitness y en el modo de utilizar la medida de fitness para crear descendientes de un par de individuos. En base a esta categorización, Grefenstette muestra que todo AG *admissible* exhibe alguna forma de *paralelismo implícito* en el sentido de que el esfuerzo de búsqueda se concentra en varias combinaciones efectivas de genes (*esquemas*) simultáneamente.

En su trabajo de 1990, Gorges-Schleuter [Gor90] ofrece una clasificación primaria de los métodos de paralelización que utilizan poblaciones no panmícticas, reconociendo tres modelos:

- El modelo de islas, con demes separados, cada uno de los cuales constituye un AG con población panmíctica.
- El modelo stepping stone, que cuenta con demes organizados de acuerdo a una estructura de vecindades no todas contiguas, lo cual limita el intercambio de material genético entre subpoblaciones.
- El modelo de vecindades, en el cual se define una relación de adyacencia sobre los individuos, los cuales pueden interactuar exclusivamente con sus vecinos.

La división entre los dos primeros modelos no es hoy considerada por los investigadores en el área, que se refieren a ambas como modelo de islas o migración.

Un resumen de los principales trabajos desarrollados en esta etapa se ofrece en la tabla a continuación.

Autor	Año	Trabajo
Holland Bossert	1960-69	Estudio de propiedades teóricas de los AG y primeras propuestas de paralelismo.
De Jong	1975	AGP Modelo maestro-esclavo.
Bethke	1976	AGP sobre una máquina paralela SIMD.
Grefenstette	1981	Primera clasificación de AGP.
Grosso	1985	Estudio de modelos de migración.
Petty et al.	1987	Implementación Parallel Genetic Algorithm (PGA)
Robertson	1987	AGP de grano fino.
Cohon et al.	1989	Equilibrios puntados y modelos multipoblación.
Tanese	1989	Estudios de efectos de parámetros de migración.
Manderick & Spiessen Mühlenbein Gorges-Schleuter	1989	AGP de grano fino.
Goldberg	1989	Modelos de AGP <i>basados en objetos</i> .
Grefenstette	1989-91	Paralelismo implícito en AG.
Gorges-Schleuter	1990	Clasificación de AGP con poblaciones no panmícticas

Tabla 1 : Principales trabajos de la etapa de investigaciones pioneras.

4 – La influencia del desarrollo de las arquitecturas paralelas

La proliferación de diseños de arquitecturas paralelas y de alta performance sobre finales de la década de 1980 y comienzo de la década de 1990 influyó positivamente en la investigación y desarrollo de versiones paralelas de los AG.

El trabajo de investigación se movió desde el terreno teórico al ámbito de las implementaciones y sus aplicaciones a la resolución de complejos problemas de optimización. Simultáneamente, el análisis comparativo de las estrategias de paralelización aplicadas se manifestó como un importante tema de estudio.

En esta etapa, que es posible considerar que se extiende hasta mediados de la década de 1990, es frecuente encontrar clasificaciones específicas y subclasificaciones para cada uno de los modelos generales identificados. En general, los investigadores centraron su trabajo en alguna arquitectura multiprocesador específica, la cual condicionaba el modelo de paralelismo a aplicar, de modo de obtener las mayores ventajas del hardware disponible.

La tendencia a las implementaciones específicas comienza a revertirse hacia el final del lustro, surgiendo trabajos con el objetivo de condensar los resultados de las investigaciones y ofrecer modelos independientes de las implementaciones y arquitecturas.

Continuando el mencionado estudio de Grosso [Gro85], Braun exploró las técnicas de migración "retrasadas" o activadas por eventos y la "degeneración" (migración que se produce luego de la convergencia absoluta) en su trabajo de 1990 [Bra90].

En su trabajo de 1991, Starkweather et al. [Sta91] estudian si es posible, y en caso de serlo bajo que condiciones, que un AGP proporcione una solución de mejor calidad que un AG serial. Los autores especulan que los AGP funcionarán adecuadamente para funciones de fitness separables, en las cuales combinando soluciones parciales es posible obtener una mejor solución, que en aquellos problemas con funciones de fitness no separables. Esta línea de trabajo será extendida más adelante por Whitley et al. [Whi97].

Como consecuencia de su simplicidad de implementación y análisis, el modelo maestro-esclavo se popularizó como el primero en implementarse sobre arquitecturas paralelas.

Fogarty y Huang [Fog91] trabajaron implementando el modelo maestro-esclavo sobre una red de transputers, concluyendo que el importante crecimiento del overhead de comunicación constituía un impedimento para futuras mejoras de sus resultados. Varios investigadores chocaron contra este problema de la escalabilidad del modelo maestro-esclavo. Al agregar procesadores, la eficiencia del AGP decrecía como consecuencia del overhead introducido para posibilitar la comunicación.

Abramson y Abela [AbA92] implementaron un AGP panmíctico sobre una arquitectura de memoria compartida (Encore Multimax de 16 procesadores) para resolver el problema de asignación de horarios escolares, reportando speedups limitados. Más adelante, trabajando sobre esta implementación, Abramson, Mills y Perkins [AMP93] agregaron una máquina de memoria distribuida (Fujitsu AP1000 con 128 procesadores) y lograron buenos speedups para ambos equipos, al trabajar con hasta 16 procesadores, degradándose la performance al incrementar el número de estos.

En su trabajo de 1992, Baluja [Bal92] resume las ideas detrás del desarrollo contemporáneo de la teoría de equilibrios puntuados, original de Cohoon [Coh87].

Esta teoría se basa en los principios de especiación alopátrica –que involucra la rápida evolución de nuevas especies en un entorno separado– y de estabilidad –ausencia de cambio. Considera que la reintroducción de viejos especímenes constituye un poderoso mecanismo para recomponer la diversidad en un ambiente que alcanzó un equilibrio, pero donde el cambio es beneficioso. Este concepto es la base de las técnicas de migración.

En su trabajo, Baluja modifica la idea de la arquitectura básica de los AGP masivamente paralelos existente hasta el momento, los que asumían que aunque las poblaciones separadas introducen un beneficio, reduciendo la severidad de los bordes entre poblaciones puede ser posible sobrellevar los problemas asociados con la introducción súbita de nuevo material genético. Un modo de conceptualizar esta idea es introducir solapamiento en las poblaciones. Esta estructura permite la transferencia gradual de información genética sin la introducción súbita de cromosomas. La porción de intercambio del AGP, formada por las secciones solapadas entre subpoblaciones, es inherente a su propia estructura.

Para el diseño de su modelo, denominado *massively distributed parallel genetic algorithm* (mpdGA), Baluja analiza los aspectos de estructura y localidad de las subpoblaciones y de los tamaños de las secciones solapadas, mencionando que debe llegarse a un compromiso entre la lentitud de propagación de buenos cromosomas para áreas de solapamiento pequeñas y la desviación de la idea de los equilibrios puntuados al crecer las áreas y acercarse a un modelo panmíctico.

La propuesta de implementación de Baluja se basa en la arquitectura paralela MassPar MP-1, modelo SIMD, compuesta de un array de elementos de procesamiento. Su trabajo reporta la obtención más eficiente de resultados de similar calidad para el conjunto de problemas de prueba –problemas de De Jong, funciones deceptivas y la maximización de la función gap– que los obtenidos mediante un AGP de poblaciones distribuidas, de implementación basada en Genitor [Whi90].

Como extensión a su trabajo sobre los AGP masivamente paralelos, en un trabajo posterior de 1993 [Bal93a], Baluja estudia diferentes topologías para la estructura de población y su performance sobre un conjunto de problemas estándar de prueba y el problema TSP, sin llegar a conclusiones definitivas sobre la existencia de una configuración óptima.

El problema de la convergencia prematura y la propuesta de una implementación para evitarlo mediante el modelo de migración sobre una arquitectura nCube es estudiado por Balakrishnan en su artículo de 1993 [Bal93b]. Los resultados de este trabajo indican la efectividad del mecanismo de paralelismo implementado para la impedir la convergencia anticipada aumentando la diversidad de la población.

Un modelo de paralelismo trivial, asignando varios procesadores a una misma instancia del problema que se resuelve concurrentemente mediante AG seriales, es propuesto por Shonkwiler, quien lo denomina IIP (independent, identical & parallel) en su trabajo de 1993 [Sho93]. Su método permite obtener interesantes resultados estadísticos, trabajar con diferentes condiciones iniciales y combinaciones de los parámetros, etc.

El modelo denominado *forkingGA* es introducido por Tsutsui & Fujimoto, en su trabajo de 1993 [TsF93], en base a la idea de utilizar multipoblaciones independientes para la solución de problemas multimodales.

El mecanismo de migración utilizado considera una población padre con modo bloqueante y poblaciones hijas con un modo de encogimiento como resultado de una creación de poblaciones. Cada población toma un rol diferente en la tarea de optimización, siendo responsable de diferentes sectores no solapados del espacio de búsqueda. El modelo es extendido por sus creadores en sus trabajos posteriores de 1994 [TsF94] y 1997 [TFG97].

En el trabajo de Gordon & Whitley [GoW93] se propone una clasificación que reconoce tres categorías, basadas en la división original de Goldberg [Gol89a] :

- AGP de población global, una versión simplificada del modelo maestro-esclavo que utiliza selección por torneo para simplificar el paralelismo, y sus versiones elitistas.
- AGP con modelo de islas y migración.
- AG masivamente paralelos o AG celulares, que asignan un número bajo de individuos por procesador o elemento de procesamiento (EP), llegando al caso extremo de asignar un individuo por EP, requiriéndose tantos EP como individuos en la población. En este modelo, cada individuo se procesa en paralelo en cada generación y el cruzamiento está limitado a un deme –o vecindad– del individuo. Usualmente la topología de conexión y la estructura de los demes es fija, y se corresponde con la topología de conexión de los EP en el supercomputador. Una descripción del modelo de AG celulares se ofrece por parte de Whitley en su trabajo [Whi93], en el cual justifica esta denominación por tratarse de un tipo particular de autómatas celulares.

Los autores realizan un estudio comparativo de diferentes modelos de AGP, incluyendo a su propia implementación, denominada Genitor. Genitor nació en 1989 [Whi89] como un AGP de población global y selección por rango, y evolucionó a un modelo de islas con migración en su versión Genitor II [Whi90].

Uno de los primeros AGP de grano fino implementados sobre una red de workstations interconectadas por una LAN se describe en el trabajo de Maruyama, Hirose y Konagaya [MHK93]. En este trabajo se analizan los efectos del sincronismo entre poblaciones distribuidas como influencia sobre la calidad de la solución obtenida y la mejora de performance comparada con el AG serial.

Las estrategias de distribución de datos para el modelo de AG *desordenados* (Messy Genetic Algorithms) son estudiadas en el trabajo de Merkle y Lamont [MeL93]. El modelo de AG desordenados es original de Goldberg [Gol90], ideado para tratar con las limitaciones de optimización de ciertas funciones devaluadas.

El algoritmo consta de una fase primordial en la cual se crea la población utilizando una estrategia de enumeración parcial, distribuyendo los bloques de construcción de soluciones y una fase de yuxtaposición en la cual se mezclan las soluciones encontradas en la fase primordial. La fase primordial usualmente domina en cuanto al tiempo de ejecución, por lo cual técnicas de paralelismo pueden aplicarse para mejorar la performance.

En el trabajo citado, Merkle y Lamont proponen diferentes estrategias de distribución para explotar el paralelismo de datos en el algoritmo de selección por torneo de la fase primordial. Conjuntamente con las estrategias propuestas (indexado de soluciones, indexado modificado, ordenamiento e intercalado de bloques de construcción y ordenamiento y asignación por bloques), los autores ofrecen una interesante descripción del algoritmo en un enfoque independiente de la arquitectura, y un análisis de los efectos del tamaño de subpoblaciones y su compatibilidad. Como conclusión de su trabajo, prueban el efecto significativo de las estrategias de distribución de datos sobre el tiempo de ejecución de la fase primordial, aunque dejan constancia de que no se mejora la calidad de la solución respecto al modelo secuencial.

En su trabajo de 1993, Munetomo, Takai & Sato [MTS93] estudian los diferentes esquemas de migración que permiten reducir comunicaciones innecesarias en los modelos de poblaciones distribuidas. El esquema definido, denominado *algoritmo de intercambio sigma* toma en cuenta la desviación estándar en la distribución del fitness de cada deme para determinar la frecuencia de intercambio de material genético.

El trabajo de Masakazu et al. [Mas93] propone un modelo de AGP con estructuras jerárquicas de subpoblaciones con un mecanismo de optimización de los parámetros de cada AG utilizando un meta-AG de nivel superior. Los parámetros de los operadores genéticos se codifican y se optimizan mediante el proceso evolutivo del meta-AG. El modelo se implementó sobre una red de workstations resultando un efectivo algoritmo de búsqueda distribuido.

En su trabajo de 1993, Bianchini y Brown [BiB93], analizan los aspectos de implementación de AGP sobre arquitecturas paralelas de memoria distribuida.

Sobre estas plataformas, las consideraciones de performance tradicionalmente condicionaban el diseño de los AGP, haciendo que las implementaciones centralizadas fueron dejadas de lado, superadas por una variada gama de implementaciones distribuidas, las cuales involucran menor número de cuellos de botella en las comunicaciones, y granularidad mayor en las subpoblaciones.

Sin embargo, los autores mencionan ciertos casos en los cuales la calidad de la búsqueda mejora al trabajar con una única población global respecto al resultado obtenido trabajando con varias subpoblaciones, en particular para hallar buenas soluciones iniciales aproximadas en pocas generaciones.

Trabajando sobre los modelos de Grefenstette [Gre81], los autores proponen una extensión a los prototipos sobre memoria distribuida, introduciendo aspectos de centralización en los algoritmos. El objetivo detrás de este enfoque, llamado "semi-distribuido" por los autores, consiste en resolver adecuadamente el delicado equilibrio entre calidad de la búsqueda y performance del algoritmo.

Por sobre la asignación de poblaciones a procesadores, se propone la organización de los procesadores en clusters, logrando un cierto grado de centralización, manteniendo el enfoque de poblaciones distribuidas sin generar un aumento excesivo en las comunicaciones. De este modo, la implementación comparte algunas de las ventajas de los modelos distribuidos y otras de los centralizados.

Basados en las ideas expuestas, Bianchini y Brown proponen una taxonomía de AGP de acuerdo con el modo de distribución de la población entre los múltiples procesadores.

Las estrategias de implementación expuestas son:

- Implementación centralizada.

Considerado como un caso extremo, consta de una única población y la implementación sigue el modelo maestro-esclavo, sobre una topología de interconexión en estrella, con el proceso maestro en el centro.

La estrategia maestro-esclavo no es práctica en la mayoría de las implementaciones sobre memoria distribuida, como consecuencia del cuello de botella en las comunicaciones hacia el maestro, la secuencialidad necesaria del proceso de replicación y la granularidad demasiado fina de los procesos que genera comunicaciones excesivas. Para aminorar el impacto de estos problemas, los autores asignan en su implementación un gran número de individuos a cada proceso esclavo, seleccionando adecuadamente los grupos de modo de que a diferentes esclavos sean asignados aproximadamente el mismo número de individuos. Complementariamente, los procesos esclavos tienen la lógica para realizar la operativa de evolución aislados por un cierto número de generaciones, antes de enviar los resultados al maestro y sincronizar con el resto de la población. El maestro continúa siendo responsable de la ejecución del mecanismo de evolución sobre la población en su conjunto, pero ésta etapa, que involucra comunicaciones, ocurre con menos frecuencia.

Para reducir el tiempo ocioso del proceso maestro mientras aguarda el resultado del trabajo de los esclavos, se le asigna una subpoblación a dicho proceso maestro, cuyo tamaño se selecciona adecuadamente de modo que el proceso maestro finalice su procesamiento antes de que los esclavos comiencen los envíos de resultados.

- Implementación semi-distribuida

Constituye un modelo intermedio entre una población global y una totalmente distribuida, que intenta evitar el cuello de botella en las comunicaciones hacia un proceso maestro. La idea consiste en organizar clusters de procesadores que trabajen como lo hace el modelo centralizado.

Los autores proponen la conexión de los procesos maestros de cada cluster en una topología toroidal, basando esta elección en cuestiones de simplicidad del algoritmo, ya que el diseño cíclico permite que cada proceso tiene un vecino en las cuatro direcciones espaciales. Cada cierto tiempo se realizan comunicaciones sincrónicas entre los maestros de los clusters, intercambiando los mejores individuos. Los esclavos se comportan del mismo modo que en la implementación centralizada.

Esta organización reduce el efecto del problema de contención causada por el acceso a un único maestro, aunque incluye cierto overhead como consecuencia de la replicación semi-distribuida y las nuevas comunicaciones entre procesos maestros.

- **Implementación distribuida**
Este modelo elimina las poblaciones compartidas. Cada procesador mantiene una porción de la población y aplica un AG secuencial sobre ella, con migración ocasional de los mejores individuos. Nuevamente se propone una topología toroidal de conexión para simplificar el algoritmo y reducir las comunicaciones. Como se reconoce en el trabajo, en las implementaciones de este estilo se asigna un esfuerzo computacional en el procesamiento de individuos poco aptos, pero este problema puede reducirse adoptando estrategias de migración adecuadas, que introduzcan la diversificación necesaria en las subpoblaciones.
- **Implementación totalmente distribuida**
Un caso extremo, que corresponde al modelo anterior, sin migración entre subpoblaciones. Es el modelo con menor cantidad de comunicaciones.

Complementando la clasificación anterior, los autores ofrecen una breve introducción a las implementaciones de AGP en arquitecturas de memoria compartida. Generalmente, los AGP centralizados se consideran más adecuados para arquitecturas de memoria compartida, ya que el mecanismo para compartir la población es proporcionado por el hardware y el SO del multiprocesador correspondiente. Disponiendo de este hardware especializado, simplemente se debe recargar el caché de los procesadores en cada generación, un mecanismo que no involucra pasaje de mensajes de ningún tipo y que se resuelve a nivel del bus de datos del equipo. Pero las limitaciones en el ancho de banda del mencionado canal y las latencias en las memorias son muy grandes comparadas con las velocidades de los procesadores, lo cual ocasiona que la performance se degrade cuando se utilizan muchos procesadores y una población numerosa.

Los autores indican que los niveles de performance alcanzados por sus versiones centralizadas y semi-distribuidas sobre arquitecturas de memoria distribuida mejoran la preformance obtenida por los enfoques centralizados sobre memoria compartida.

En su posterior extensión al trabajo anterior, Bianchini et al. [Bia95] analizan cuantitativamente las ventajas del modelo semi-distribuido sobre las restantes implementaciones. En este nuevo artículo, los autores extienden el conjunto de problemas de testeo (originalmente compuesto exclusivamente por el problema 0-1 ILP), contemplando funciones bien conocidas en los benchmarks de optimización, los cinco problemas de minimización de DeJong, el TSP y el 0-1 ILP. La conclusión de su trabajo indica que realizar selecciones centralizadas cada cierto tiempo consiste en una mejora sobre los enfoques tradicionales de distribución de población. Califican la estrategia de centralizaciones como un mecanismo extremadamente robusto que permite lograr mejoras en la calidad de los resultados obtenidos, manteniendo los tiempos de ejecución y las comunicaciones bajo control.

En su reseña de 1994, Adamidis [Ada94a] plantea una categorización de AGP basada en dos enfoques distintivos respecto a la operativa del mecanismo evolutivo, el modelo estándar y el modelo de descomposición.

El modelo estándar de Adamidis consiste en la implementación del mecanismo de los AG secuenciales sobre máquinas paralelas, correspondiendo al modelo de paralelización global o modelo maestro-esclavo. El mecanismo algorítmico del AG serial no se modifica al distribuir la evaluación de la función de fitness.

El modelo de descomposición corresponde a una operativa diferente a la de los AG seriales, donde la población existe en una forma distribuida, ya sea en forma de una única población efectivamente distribuida entre procesos o poblaciones múltiples independientes. El autor refina este modelo, considerando los AGP de grano fino (población única) y los AGP de grano grueso (modelo de islas y stepping stone).

Complementariamente, menciona la posibilidad de crear algoritmos híbridos que combinen los enfoques, los cuales cuentan con un grado de complejidad mayor que los AGP mencionados. En esta línea de investigación, el autor propuso un modelo cooperativo (CoPDEB – *Co-operating Populations with Different Evolution Behavior*) en su trabajo con Petridis de 1996 [AdP96] y en su tesis de doctorado [Ada97]. El comportamiento diferenciado de las subpoblaciones en el mencionado modelo se consigue utilizando diferentes mecanismos de selección, operadores y métodos de comunicación.

Los trabajos de Levine [Lev93], [Lev94] presentan una implementación paralela del modelo de AG de *estado estacionario*. Desde el punto de vista del porcentaje de población reemplazada en cada generación, es posible clasificar los AG en *generacionales*, aquellos que reemplazan una significativa parte de la población luego de los pareos, y AG de *estado estacionario*, que reemplazan solamente una pequeña parte de la población, usualmente un conjunto muy pequeño de individuos.

El diseño de Levine consiste en un modelo de islas con poblaciones múltiples independientes que evolucionan de acuerdo a un AG de estado estacionario y pequeñas migraciones ocasionales de los individuos más aptos.

La propuesta de AGP de estado estacionario fue implementada en un IBM SP-1 con 128 nodos y aplicado al problema de particionamiento de conjuntos para modelar el control de tráfico aéreo, mostrando significativos resultados en términos de mejora de performance y calidad de la solución hallada. Otro resultado destacable lo constituye la capacidad de escalabilidad de la implementación para resolver problemas más complejos.

En la clasificación de implementaciones de AGP ofrecida en su trabajo de 1994, Lin et al. [Lin94] describen las tres categorías clásicas ya comentadas. Su trabajo se especializa en los AGP de grano grueso, analizando diferentes estrategias de migración y esquemas de conectividad, como principal mecanismo para resolver el problema de la convergencia prematura de los AG seriales o de población única. Los autores resaltan la capacidad de los AGP de mantener poblaciones múltiples, capaces de evolucionar independientemente, permitiendo una mejor y más exhaustiva búsqueda en el espacio de soluciones.

Las categorías de AGP de su clasificación son:

- AGP micrograno
Modelo de población única, con distribución tipo maestro-esclavo de evaluación de la función de fitness. No resuelve el problema de convergencia prematura, pero logra mejoras significativas de performance respecto al AG serial para problemas complejos.
- AGP de grano fino
Modelo que asigna un individuo a cada elemento de procesamiento. Cada individuo forma parte de varias subpoblaciones, solapadas entre sí.
La adyacencia entre individuos está determinada por la topología de interconexión entre los elementos de procesamiento. La alta conectividad entre vecinos incrementa la difusión de individuos con mejor fitness, pero hace a la población susceptible al dominio de algún esquema de codificación y podría llevar a convergencia prematura. Limitar las interacciones, por su parte, reduce la performance de la implementación. Los autores plantean que en las implementaciones de este modelo debe lograrse un compromiso entre estos factores para obtener una solución de calidad, con buena performance.
- AGP de grano grueso
Modelo caracterizado por poseer una gran cantidad de individuos asignados a poblaciones semi-independientes.
Los autores estudian los diferentes criterios que permiten clasificar a los AGP de grano grueso:
 - El modelo de migración, que determina cómo y cuando se realiza el intercambio de individuos, diferenciando entre modelo aislado, donde las poblaciones evolucionan independientemente sin existir comunicación, el modelo de islas sincrónico, y el modelo de islas asincrónico, en los cuales la migración debe ocurrir en un punto común para las subpoblaciones (sincrónico) o puede ocurrir en eventos no relacionados con la evolución de las subpoblaciones (asincrónico)
 - El esquema de conectividad entre poblaciones, que puede ser estático, basado en las topologías de los multicomputadores de alta performance, o dinámico, un enfoque en el cual la topología de conexión varía durante la evolución.
 - El criterio de homogeneidad de las poblaciones. Un AGP homogéneo utiliza los mismos mecanismos de codificación, selección, cruzamiento y reemplazo, e inclusive los mismos criterios para la selección de sus parámetros. Por su parte, los AGP no homogéneos permiten a las subpoblaciones utilizar diferentes mecanismos evolutivos, con criterios propios para la selección de parámetros, e inclusive utilizando codificaciones diferentes para las soluciones del problema. La variación de representaciones y esquemas evolutivos puede resultar beneficiosa para problemas de optimización multiobjetivo o con espacios de búsqueda extensos.

Lin et al. proponen un modelo de AGP asincrónico, estático (característica no limitante del modelo, que podría ser dinámico) y heterogéneo, denominado AG Injection Island (AGII). Este algoritmo tiene como característica distintiva su heterogeneidad, utilizando diferentes resoluciones (largos de strings de representación) para las codificaciones de los individuos en cada población.

El mecanismo de migración es solo de "una vía", el intercambio se da exclusivamente desde poblaciones de baja resolución a poblaciones de alta resolución. Este modelo de migración requiere de un mecanismo "traducción" entre representaciones, la cual se hace sin pérdida de información al considerar la jerarquía de intercambio indicada. Dicha jerarquía establece un mecanismo de "refinamiento" de soluciones, basado en las reglas de migración, que comprenden cuatro modelos (simple, completa, estricta, pobre) diferenciados por los diferentes destinos de los individuos migrados.

El AGII realiza la búsqueda en múltiples codificaciones, cada una concentrada en diferentes áreas del espacio de soluciones. El espacio de búsqueda para nodos de baja resolución es proporcionalmente menor, lo cual permite hallar soluciones "adecuadas" rápidamente, que son inyectadas a los nodos de alta resolución para su posterior refinamiento.

La jerarquía define nodos padre e hijo, que comparten porciones del mismo espacio de búsqueda. La búsqueda de rápidos candidatos en baja resolución por parte del padre puede potencialmente orientar al hijo en su búsqueda.

Este enfoque permite una aplicación directa de una técnica Divide & Conquer no garantida por los AGP homogéneos, que en general producen soluciones pertenecientes a varios subespacios de búsqueda.

Conjuntamente, los nodos con el tamaño de bloque más pequeño pueden hallar soluciones con la mayor resolución. Los AGII procesan los diferentes niveles de resolución en paralelo y reducen el riesgo de búsqueda en intervalos incorrectos evitando el fenómeno de convergencia prematura.

En su completo trabajo de 1996, Schwehm [Sch96] discute los tipos de AGP independizándose de aquellas implementaciones diseñadas sobre hardware especial.

El resultado es una categorización precisa de los AGP que describe claramente las características relevantes de cada modelo.

Analizando el modelo de poblaciones paralelas, Schwehm propone una clasificación basada en la organización estructural de la población. Las categorías propuestas, de menor a mayor grado de estructuración, corresponden a modelo global (población no estructurada y mecanismo de selección global), modelo regional (subpoblaciones y modelo de selección local) y modelo local (población provista con una estructura de vecindad, que define el mecanismo de selección de individuos para el cruzamiento).

Luego de exponer su clasificación, explica la correspondencia existente con los modelos de otras taxonomías, algunas de ellas estudiadas en esta reseña. ([Gre81], [Man89], [Gor92], [Dor93], [CaP95]).

Las categorías de la clasificación de Schwehm se comentan a continuación:

- El modelo global implementa el AG clásico sobre una plataforma paralelo-distribuida. Se identifica al mecanismo de selección proporcional como el cuello de botella de la implementación, mencionando el análisis de Bethke [Bet76]. Dentro del modelo global, es posible identificar variantes de acuerdo al mecanismo para lidiar con dicho problema. El autor menciona a los modelos globales con selección secuencial, que omiten la paralelización del mecanismo de selección. Este tipo de AGP son implementables como un típico modelo maestro-esclavo con la población almacenada en memoria compartida o en el proceso maestro, y son útiles cuando la evaluación de la función de fitness es costosa. Asimismo se mencionan los modelos globales con selección paralela, que distribuyen el cálculo del fitness absoluto para el mecanismo de selección proporcional, o el mecanismo de ordenación para mecanismo de selección por rangos, ambos con costo logarítmico en el número de individuos; o el mecanismo de selección por torneo, más simple de paralelizar. En estos enfoques subsiste el problema de obtener los individuos para el cruzamiento una vez seleccionados, sin conflictos de memoria o cuellos de botella en las comunicaciones.

- El modelo regional constituye una extensión al modelo clásico de los AG. La población se divide en subpoblaciones (regiones o demes) cada una de las cuales ejecuta un AG independientemente. Los mecanismos globales de selección, cruzamiento y reemplazo se reemplazan por mecanismos locales y se agrega un operador de intercambio de material genético entre las poblaciones (migración).
Dentro de este modelo, diversas variantes pueden establecerse de acuerdo a la estructura y número de población proporcionada a las regiones, así como las topologías de migración y los parámetros que la controlan –estrategia de migración, tasa de migración, frecuencia de migración-. Mediante la modificación de estos parámetros, una determinada instancia del modelo regional puede acercarse más al modelo global –utilizando una topología de migración completa, alta tasa y frecuencia de migración- o al modelo local – utilizando una topología de migración rígida.
- Por último, el modelo local consiste en una modificación de la estructura de los AG, en la cual se provee a la población de una estructura espacial. Los mecanismos de selección, reproducción y reemplazo se llevan a cabo localmente, dentro de una vecindad definida por proximidad espacial entre los individuos. La estructura de la vecindad determina la ubicación espacial de los individuos y los caminos de propagación de material genético entre la población. La topología de la vecindad y su tamaño (radio) consisten en los principales parámetros espaciales de este modelo. Las estrategias de selección local y reemplazo local determinan la presión de selección, usualmente menor en este modelo como consecuencia de las limitaciones impuestas por la estructura espacial. Las estrategias más comunes de cruzamiento local utilizan el valor de fitness proporcional local o un rango proporcional, e inclusive la selección por torneo local. Pueden seleccionarse dos individuos de la vecindad para reproducción o seleccionarse un único individuo que forzosamente se apareará con el individuo central de la vecindad. La estrategia de reemplazo estándar para este modelo es el reemplazo generacional sustituyendo el individuo que ocupa el lugar central de la vecindad, ya que el reemplazo elitista no puede utilizarse como técnica local, al necesitar conocimiento de datos de la población global.

Es posible considerar este artículo como la culminación de la segunda etapa en la evolución del diseño y las clasificaciones de AGP, por ofrecer una taxonomía completa, independiente de las implementaciones particulares y del hardware subyacente, y la correspondencia existente con los modelos de otras clasificaciones de AGP.

A modo de resumen, los principales trabajos de la segunda etapa en la evolución del diseño y las clasificaciones de AGP se ofrecen en la tabla 2.

Autor	Año	Línea de trabajo
Braun	1990	Técnicas de migración "retrasadas"
Starkweather et al.	1991	Condiciones para que un AGP supere calidad de un AG.
Fogarty y Huang	1991	Modelo maestro-esclavo sobre una red de transputers.
Abramson y Abela	1992	AGP panmíctico sobre memoria compartida.
Baluja	1992	AGP masivamente paralelos.
Shonkwiler	1993	Modelo de paralelismo trivial, sin comunicación.
Gordon & Whitley	1993	Clasificación de AGP sobre modelos de Goldberg.
Maruyama et al.	1993	AGP de grano fino implementado sobre una red de workstations interconectadas por una LAN.
Merkle y Lamont.	1993	estrategias de distribución de datos para el modelo de AG <i>desordenados</i> (<i>Messy GA</i>).
Tsutsui & Fujimoto	1993	ForkingGA – multipoblaciones independientes para la solución de problemas multimodales.
Masakazu et al.	1993	modelo de AGP con estructuras jerárquicas.
Munetomo, Takai, Sato	1993	Mecanismos de migración para reducir comunicaciones.
Bianchini y Brown	1993	AGP sobre arquitecturas de memoria distribuida.
Adamidis	1994	Categorización de AGP basada en la operativa del mecanismo evolutivo.
Levine	1994	Modelos de AGP <i>de estado estacionario</i> .
Lin et al.	1994	Clasificación de AGP, especializada en grano grueso.
Schwehm	1996	Clasificación de AGP independiente del hardware.

Tabla 2 : Principales trabajos de la etapa influida por la difusión de arquitecturas paralelas.

5 – La generalización y unificación de los modelos

La tercera etapa en la evolución de las clasificaciones de AGP se extiende desde mediados de la década de 1990 hasta nuestros días. Los trabajos de investigación en esta etapa se caracterizan por tratar de independizarse totalmente del hardware subyacente, frecuentemente analizando implementaciones de diversos modelos y tratando de ofrecer taxonomías genéricas, que contemplen la mayoría de los modelos existentes.

El estudio de los modelos distribuidos domina esta etapa, motivado por de la difusión de las arquitecturas paralelas distribuidas de bajo costo (clusters de PCs o workstations), aunque continúan desarrollándose implementaciones para supercomputadores de memoria compartida.

Tomassini, en su reseña sobre AG de 1995 [Tom95], y en su reseña posterior, específica sobre AGP [Tom99], hace referencia a varios modelos de programación paralela aplicada a los AG. Siguiendo el modelo de Schwehm ya comentado [Sch96], el autor reconoce la existencia de jerarquías de aplicación del paralelismo.

La aplicación del paralelismo a nivel de la función de fitness conduce al modelo maestro-esclavo, que Tomassini denomina modelo de *paralelismo global*.

La aplicación del paralelismo a nivel de la población permite implementar el modelo de islas, constituido por poblaciones semi-independientes que implementan el AG serial, con el agregado de migraciones de individuos. Tomassini aún considera la versión *stepping stone*, donde la migración solamente ocurre entre subpoblaciones vecinas.

Cuando el paralelismo se aplica a nivel de individuos, se obtienen los AG *celulares* o de grano fino. La terminología *celular* es adoptada por su similitud con los autómatas celulares con reglas de transición estocásticas, de acuerdo al trabajo de Whitley ya citado [Whi93] y a trabajos propios [Tom93a] [Tom93b]. Menciona la adecuación de este modelo a las arquitecturas SIMD MasPar y CM200, que proporcionan comunicación eficiente implementada por hardware.

Tomassini reconoce para cada modelo las versiones sincrónicas y asincrónicas, dependiendo del mecanismo de coordinación entre poblaciones mediante las comunicaciones, ofreciendo una clasificación dependiendo del comportamiento espacial y temporal del AGP, la cual se resume en la tabla que se ofrece a continuación:

	Grano Grueso	Grano Fino	
Sincronismo/Nivel	Población	Individuo	Fitness
Sincrónico	Islas Sincrónicas	Celular	Maestro-esclavo
Asincrónico	Islas Asincrónica	Estocástico	Maestro-esclavo asincrónico.

Tabla 3 : Modelos de la clasificación de Tomassini

Fuera de su clasificación, Tomassini coloca a los AGP híbridos –que combinan enfoques de paralelismo–, y a los AGP no uniformes –cuyas poblaciones utilizan diferentes codificaciones, mecanismos evolutivos y/o parámetros–, englobados dentro de una categoría de modelos de AGP *No estándar*.

Simultáneamente al trabajo de Adamidis [AdP96], [Ada97], Venkateswaran, Obradovic y Raghavendra en su trabajo de 1999 [VOR96] proponen independientemente un modelo de AG *cooperativos*, que corresponde a un enfoque intermedio entre el modelo global y el modelo de islas. Múltiples procesos ejecutan un AG global en paralelo, cada uno con la totalidad de la población, trabajando sobre diferentes áreas del espacio de soluciones, lo cual mejora las probabilidades de obtener mejores soluciones. El modelo es heterogéneo, en el sentido de que cada proceso cuenta con sus propios mecanismos de selección y reemplazo, operadores de cruzamiento y mutación. Esta característica hace al algoritmo menos sensitivo a una determinada elección de estos parámetros. Adicionalmente, existe cooperación entre los procesos, que intercambian individuos periódicamente con el objetivo de resolver el problema eficientemente.

El modelo fue implementado de acuerdo a un modelo maestro-esclavo sobre una plataforma distribuida (red de computadores) y aplicado a la resolución de problemas de asignación, reportándose resultados óptimos para pequeños problemas y mejoras significativas de performance respecto al modelo serial para problemas donde la relación entre tiempo de comunicación y tiempo de ejecución es pequeña.

En sus trabajos [Whi97], [Whi98a], [Whi98b], [Whi99] Whitley et al. realizan un exhaustivo análisis del modelo de islas con subpoblaciones y migración. En estos trabajos se estudian detalladamente los parámetros que controlan la operación de migración, el intervalo de migración y el tamaño de la población migrada. Conjuntamente, se analizan diferentes representaciones para problemas de optimización y las mejoras obtenidas con respecto a la calidad de la solución al utilizar el modelo de islas especialmente para problemas linealmente separables. Sus resultados muestran la capacidad de los algoritmos con multipoblaciones semi-independientes para resolver este tipo de problemas. Aún para problemas con funciones no separables, los resultados indican que el uso de migración introduce una mejora de performance significativa respecto al modelo panmítico.

Los trabajos en el área por parte de Cantú-Paz [CaP98a], [CaP98b], [CaP99], culminaron con la publicación de un completo texto que reúne análisis teóricos y aspectos de implementación y estudios de eficiencia para AGP [CaP00c].

La clasificación de AGP de Cantú-Paz extiende los modelos clásicos de Grefenstette [Gre81], coincidiendo con las propuestas de Adamidis [Ada94a], Lin et al. [Lin94], Alba y Troya [AIT99a] y Tomassini [Tom99]. Tomando en cuenta la distribución de la población y de las operaciones de evolución, reconoce las siguientes categorías:

- AGP modelo maestro-esclavo
- AGP de población múltiple.
- AGP de grano fino.
- AGP paralelos híbridos.

Cantú-Paz estudia exhaustivamente los AGP modelo maestro-esclavo de población única en su trabajo de 1998 [CaP98b]. El modelo se basa en una sencilla aplicación del paralelismo a los AG, que consiste en distribuir la evaluación de la función de fitness entre un conjunto de procesos esclavos, manteniendo las operaciones del AG (selección, cruzamiento y mutación) en un único proceso maestro.

El autor destaca que el enfoque cuenta con ciertas ventajas respecto a otros modelos de AGP: utilizan el mismo mecanismo de exploración del espacio de soluciones que los AG seriales, lo que hace aplicables directamente las consideraciones de diseño deducidas teórica y experimentalmente para éstos; la sencillez de su implementación, que los popularizó especialmente durante las primeras etapas del desarrollo de AGP y las mejoras de performance significativas al resolver problemas con funciones de fitness complejas.

En una primera etapa, Cantú-Paz estudia el modelo sincrónico de AGP maestro-esclavo, para obtener cotas inferiores simples para la eficiencia del modelo, ya que este enfoque introduce puntos de sincronización antes de comenzar a procesar la siguiente generación.

De este modo, analiza el tiempo de ejecución, identificando sus componentes de cálculo y de comunicación de información entre procesos, obteniendo resultados para el número óptimo de procesadores, el speedup y la eficiencia para el modelo sincrónico. El tiempo de cálculo requerido por el proceso evolutivo depende del tamaño de la población, que no puede reducirse arbitrariamente sin una pérdida potencial de la calidad de la solución. El tiempo de evaluación de la función de fitness puede disminuirse aumentando el número de procesos esclavos, de modo de asignar menor cantidad de individuos a cada uno de ellos.

El tiempo de comunicación depende del número de procesos esclavos utilizado, y el overhead de comunicación será mayor al aumentar la cantidad de procesos.

Cantú-Paz argumenta que la eficiencia de las implementaciones maestro-esclavo crece cuando las evaluaciones son más costosas, logrando alcanzar speedups casi lineales para problemas complejos. Cita como ejemplos los trabajos de Grefenstette de entrenamiento de robots [Gre95], y de Punch et al. [Pun93] en selección y clasificación de características.

De acuerdo a esta idea, es posible identificar un problema adecuado para resolver con este modelo midiendo el tiempo de evaluación serial de la función de fitness y comparándolo con el tiempo requerido por las operaciones de evolución, cuya paralelización de las operaciones de evolución no es usual ni útil, ya que necesitan información global y la mejora de performance se pierde en tareas de comunicación y sincronización.

En su análisis del modelo maestro-esclavo sincrónico, Cantú-Paz muestra que la eficiencia llega al 50% en la configuración óptima, lo cual indica que aproximadamente la mitad del tiempo los procesadores se encuentran ociosos, esperando que el maestro envíe datos para procesar. Sin embargo, en el modelo sincrónico el maestro no puede enviar nuevos datos hasta que todos los esclavos hayan finalizado su tarea. La sincronización es necesaria si se desea que el comportamiento del AG paralelo actúe exactamente como lo haría el serial, pero es posible evitarlo si se está dispuesto a aceptar que el AG paralelo se comporte diferente al serial.

En el modelo maestro-esclavo asincrónico, tan pronto como un esclavo finaliza su tarea envía su resultado al maestro, quien inserta los individuos cuya función fue evaluada en la población. Luego, el maestro genera nuevos individuos, y los envía a los esclavos disponibles. Este modelo tiene el potencial de ser más eficiente que el sincrónico, en especial si los tiempos de evaluación de la función de fitness no son constantes para todos los individuos. Como contrapartida, la complejidad algorítmica de este modelo es mayor, ya que se introducen nuevos parámetros que controlan el comportamiento del AG.

Cantú-Paz vincula los AGP maestro-esclavo asincrónicos con el modelo sincrónico con salto generacional de De Jong, ya descritos [DeJ75]. La diferencia entre estos modelos consiste en la manera en la cual se incorporan los nuevos individuos a la población, asunto que debe resolverse en cada uno de los modelos maestro-esclavo. Algunas de las alternativas consisten en insertar los nuevos individuos aleatoriamente, reemplazar a los peores individuos de la generación anterior, o competir con otros individuos para ganar un lugar en la población.

Si la inserción de individuos se basa en sus valores de fitness, la presión de selección se incrementará, y el AG podría necesitar mayores poblaciones para obtener una solución de calidad similar a la del algoritmo con menor presión.

El paralelismo asincrónico puede tener otros efectos sobre el AG, además de incrementar la presión de la selección. El más notable es que los individuos pueden retornar de los esclavos en un orden diferente al que fueron creados, al finalizar ciertos esclavos sus evaluaciones antes que otros. Aguardar por los individuos ordenadamente es posible, pero en este caso decrece la performance.

Cantú-Paz cita el estudio empírico de Davison y Rasheed [DaR99], que indica que el efecto de este mecanismo aleatorio no es significativo en el funcionamiento del AG, aún para muchos procesos esclavos, del orden del tamaño de la población.

Los AGP maestro-esclavo asincrónicos son algo más complejos de implementar, pero la ganancia en performance supera el pequeño esfuerzo adicional. Sin embargo, el no determinismo de los procesos asincrónicos hacen a este tipo de AGP más difíciles de analizar que su modelo sincrónico.

Si bien el modelo maestro-esclavo no parece intuitivamente ser la mejor opción de diseño para un AGP de población distribuida tipo de AG paralelos, Cantú-Paz lo analiza para obtener una cota inferior a la performance de los AGP con poblaciones múltiples distribuidas.

En su línea de trabajo de aplicación del paralelismo a los AG desordenados, Lamont, Gates & Merkle proponen una implementación utilizando la biblioteca de desarrollo de aplicaciones paralelo-distribuidas MPI en su trabajo de 1997 [LGM97]. El modelo, basado en el algoritmo fmGA (fast messy genetic algorithm) de Goldberg et al. [Gol93], está concebido para explotar las características de cálculo local y global de los AG desordenados. La escalabilidad del modelo propuesto se estudia en el trabajo de los mismos autores de 1998 [LGM98].

En su trabajo de 1999, Alba y Troya [Alb99a] analizan la importancia del sincronismo en las migraciones para AGP con poblaciones distribuidas. Para su trabajo, identifican a los AGP de grano fino y de grano grueso como subclases de un único modelo de AGP formado por un conjunto de subalgoritmos comunicados. Proponen la denominación de AG *distribuidos* y *celulares* para los modelos de grano grueso y fino respectivamente, considerando que la granularidad de un problema paralelo refiere a la relación entre trabajo computacional y comunicación, mientras que en el caso de los modelos de AGP se diferencian en el modo que estructuran la población. Los modelos *distribuidos* cuentan con subpoblaciones numerosas y subalgoritmos poco relacionados, mientras que los modelos *celulares* cuentan con pocos subalgoritmos fuertemente relacionados, que operan con pocos individuos.

Como resultado de sus trabajos donde efectúan comparaciones entre modelos distribuidos y celulares sincrónicos y asincrónicos, [ACT99b], [ACT99b], [AIT99b] [AIT01], Alba y Troya confirman que los segundos superan en término de performance confirmando resultados similares de otros autores.

La investigación de implementaciones de AGP para plataformas específicas disminuye en comparación con el período anterior, pero no desaparece.

En este sentido, Baraglia y Perego estudian en su trabajo de 1999 [BaP99] la implementación de AGP de grano fino sobre multiprocesadores con conexión de hipercubo. En su breve taxonomía discriminan entre el modelo *centralizado* con población panmíctica y los modelos distribuidos de *grano fino* y de *grano grueso*.

Los autores ofrecen un análisis comparativo entre AGP de grano fino y de grano grueso implementados sobre un multiprocesador nCUBE 2 de 128 nodos para la resolución del TSP, mostrando la potencial ventaja de escalabilidad de la implementación de grano fino al crecer el número de procesadores.

Un original modelo híbrido de poblaciones distribuidas con secciones compartidas es presentado en el trabajo de Hiroyasu, Miki y Watanabe de 1999 [HMW99]. El diseño peculiar del modelo permite lograr diversidad y alta precisión en las soluciones, características necesarias para la resolución de problemas de optimización multiobjetivo.

En su artículo de 1999, Sprave [Spr99] definió un modelo formal unificado para las estructuras de población no panmícticas en algoritmos evolutivos. El objetivo de su trabajo consiste en definir una propuesta para la terminología y el modelo matemático de las estructuras multipoblacionales, hasta el momento dominadas por términos y modelos análogos a los del modelo secuencial o vinculados con la computación de alta performance. Sprave propone modelar las estructuras de población mediante hipergrafos, estudiando los modelos de migración y de grano fino. El entorno definido por Sprave se presenta como una poderosa herramienta para el cálculo de medidas de la presión de selección, entre las cuales el trabajo de Sprave incluye el cálculo del diámetro de la estructura de población y de la *probabilidad de posesión (takeover probability)*, vinculada con el concepto de *tiempo de posesión (takeover time)*, que mide el número de generaciones necesarias para que el mejor individuo de la población inicial saturar la totalidad de la población, asumiendo que el único operador aplicado es el de selección.

Los efectos del paralelismo global aplicado a los AG de estado estacionario se describen en el trabajo de Davidson y Rasheed [DaR99].

El resultado principal de la investigación empírica con el algoritmo GADO (Genetic Algorithm for Design Optimization [Ras98]) prueba que el modelo de paralelismo maestro-esclavo permite obtener mejoras de performance significativas sin afectar la convergencia del mecanismo evolutivo del AG serial.

El paralelismo aplicado a los AG de estado estacionario también se ha desarrollado en los últimos años. En este sentido se destaca el trabajo de Berger, Barkaoui & Bräysy [BBB01] quienes presentan un modelo con poblaciones solapadas aplicado a la resolución del problema de Ruteo de Vehículos con Ventanas de Tiempo.

El modelo presentado (denominado *PHGA*) combina la potencialidad de la evolución paralela y la relajación parcial con constantes temporales para lograr una exploración exhaustiva del espacio de búsqueda.

El trabajo de este grupo de investigadores incluye innovaciones a los operadores de evolución del AG estándar, para incorporar conceptos emergentes de técnicas de optimización promisorias como las heurísticas de inserción o las colonias de hormigas. De acuerdo a los resultados presentados, el modelo ha alcanzado su objetivo de lograr una y mejora de la intensidad y de la diversificación de la búsqueda. El modelo *PHGA* se reporta como un algoritmo competitivo y de mejor performance que las metaheurísticas conocidas para los problemas de Ruteo de Vehículos con Ventanas de Tiempo.

En su trabajo de 1999, Nowostawski & Poli [NoP99b] presentan una taxonomía unificada que reúne un amplio número de modelos propuestos en la literatura. Su trabajo contempla varias categorías, siguiendo un enfoque globalizador, para tomar en cuenta la mayoría de los modelos descritos en este trabajo, en un intento de que su propuesta comprenda a la totalidad de las clasificaciones elaboradas hasta ese momento.

Al número de ocho ascienden las categorías identificadas por Nowostawski y Poli, quienes distinguen entre los siguientes modelos :

- Modelo maestro-esclavo, con sus versiones sincrónicas y asincrónicas.
- Modelo distribuido de subpoblaciones con migración.
- Modelo de poblaciones estáticas solapadas, sin migración.
- AG masivamente paralelos.
- Modelo de poblaciones dinámicas, con solapamiento.
- AGP de estado estacionario.
- AG desordenados paralelos.
- Modelos híbridos.

El modelo monopoblación maestro-esclavo, incluye sus versiones sincrónicas y asincrónicas, de acuerdo al criterio adoptado para coordinar las poblaciones antes de iniciar una nueva generación.

El modelo distribuido de subpoblaciones con migración es considerado uno de los enfoques más difundidos de aplicación de paralelismo a los AG. El nuevo operador de migración caracteriza a este modelo, y permite discriminar entre sus diferentes variantes de acuerdo a sus parámetros, topología de migración, tasa de migración, frecuencia de migración y esquema de selección y reemplazo de individuos migrados. En su clasificación, los autores mencionan al ya extinto modelo *stepping stone* como un caso particular de este modelo. En los extremos de este enfoque se posicionan los AGP de grano fino y de grano grueso, de acuerdo a la cantidad de subpoblaciones y los individuos asignados a ellas. Los AGP de grano grueso son usualmente implementados en modelos de máquinas paralelas MIMD distribuidas, mientras que los AGP de grano fino son adecuados para arquitecturas SIMD de memoria compartida.

El principal inconveniente de este modelo consiste en la dificultad de implementar soluciones escalables para grandes problemas, lo cual lleva generalmente a diseñar modelos híbridos que permitan un uso más inteligente del hardware paralelo disponible, introduciendo jerarquías de procesamiento.

El modelo de poblaciones estáticas solapadas es similar al anterior, pero no cuenta con un operador de migración. El intercambio de información entre las subpoblaciones se realiza a través de áreas de solapamiento, formadas por individuos que pertenecen a más de un deme. La ausencia de comunicaciones explícitas hace a este modelo adecuado para su implementación en arquitecturas paralelas SIMD, realizando los intercambios a través de memoria compartida.

Los AG masivamente paralelos son una extensión del modelo anterior, con un número muy elevado de demes y pocos individuos –idealmente solo uno– asignado a cada deme. El mecanismo de intercambio de material genético en este modelo se conoce como *difusión*, y se encuentra limitado por la estructura subyacente a la población, usualmente en forma de grilla. Este modelo es adecuado para computadores masivamente paralelos que proporcionen mecanismos eficientes de comunicación entre elementos de procesamiento.

El modelo de poblaciones dinámicas con solapamiento es un aporte de los propios Nowostawski y Poli. Los orígenes del modelo pueden encontrarse en los trabajos previos de Nowostawski, [KwN97], [Now98a] y [Now98b], siendo definido y descrito en detalle en su trabajo [NoP99b]. Surge como una propuesta híbrida que combina la evaluación distribuida de fitness del modelo maestro-esclavo y el modelo de subpoblaciones estáticas con solapamiento. En cada generación, los demes son creados dinámicamente, siendo este mecanismo de reorganización el encargado del intercambio de información entre subpoblaciones. La principal ventaja de la introducción de la reorganización de los demes es eliminar el efecto de los individuos más lentos en el modelo maestro-esclavo. Cada deme se crea tan pronto como existan suficientes individuos para ello, y en ese momento se inicia la nueva generación del AG.

Desde el punto de vista del procesamiento paralelo, corresponde a un algoritmo MIMD maestro-esclavo asincrónico totalmente escalable, adecuado para la ejecución en máquinas distribuidas o masivamente paralelas.

Los AG de estado estacionario, por oposición a los AG generacionales, se basan en generar en cada iteración del algoritmo unos pocos individuos nuevos, los cuales son reinsertados en la población de acuerdo al mecanismo de reemplazo elegido ([Whi98a], [Dav91]) Este mecanismo de actualización continua de la población, posibilita una aplicación sencilla de las técnicas de paralelismo. La aplicación de los mecanismos de selección y reemplazo debe efectuarse secuencialmente al encontrarse los individuos en la sección crítica del algoritmo, mientras los restantes operadores pueden realizarse en paralelo.

Los autores incluyen una categoría para los AG desordenados paralelos. Las características de este modelo de AG permiten aplicar técnicas específicas de paralelismo, las cuales fueron comentadas en este documento, al analizar el trabajo de Merkle y Lamont [MeL93]

Los modelos híbridos combinan algunos de los diferentes enfoques expuestos para paralelizar los AG. Principalmente definen jerarquías de ordenamiento entre poblaciones, aplicándose en cada una de las jerarquías enfoques de paralelización diferentes, con el objetivo de combinar las ventajas de dos o más modelos de paralelismo. Como ejemplo, es común combinar múltiples poblaciones con el modelo maestro-esclavo o con el modelo de grano fino. Su estructura los hace adecuados para implementar sobre multiprocesadores. Los modelos híbridos son estudiados en detalle en el texto de Cantú-Paz [CaP00c], bajo el nombre de *modelos jerárquicos*.

Los trabajos relevantes de la tercer etapa en la evolución del diseño y las clasificaciones de AGP se ofrecen en la tabla 4.

Autor	Año	Línea de trabajo
Tomassini	1995 1999	Clasificación de AGP
Adamidis. Venkateswaran, Obradovic, Raghavendra	1996	Modelo de AG <i>cooperativos</i> .
Whitley et al.	1997 1998 1999	Análisis del modelo de islas con subpoblaciones y migración
Cantú-Paz	1998 1999 2000	Diseño e implementación de AGP.
Lamont, Gates, Merkle	1997	Paralelismo para AG desordenados, usando MPI.
Alba y Troya	1999	Importancia del sincronismo en las migraciones para AGP con poblaciones distribuidas.
Baraglia y Perego	1999	AGP de grano fino sobre multiprocesadores con conexión de hipercubo
Hiroyasu,Miki,Watanabe	1999	AGP híbrido de poblaciones distribuidas con secciones compartidas
Davidson y Rasheed	1999	Paralelismo global aplicado a AG de estado estacionario
Sprave	1999	Modelo formal unificado para las estructuras de población no panmícticas en algoritmos evolutivos.
Nowostawski & Poli	1999	Taxonomía unificada de AGP.
Berger, Barkaoui,Bräysy	2001	Modelo con poblaciones solapadas para AG de estado estacionario

Tabla 4 : Principales trabajos de la etapa de generalización y unificación de los modelos .

Capítulo 6 - Panorama del estado actual

El trabajo desarrollado en el área en los últimos años es extenso, y el esfuerzo por lograr un modelo unificado para categorizar las diferentes maneras de aplicar las técnicas de procesamiento paralelo a los AG puede notarse en los comentarios del capítulo precedente. Sin duda, la referencia inmediata al estado actual de investigación en el área de los AGP lo constituye el texto de Cantú-Paz [CaP00c]. Sin embargo, la clasificación utilizada por el mencionado autor no contempla algunos enfoques de paralelismo que poseen características intrínsecas que los diferencian de los modelos estándar discriminados en el referido texto. Complementariamente, la categorización de *modelos híbridos*, aunque extensamente analizada por Cantú-Paz, reúne una amplia gama de enfoques, y es susceptible de un refinamiento.

Por los motivos expuestos, la clasificación de Nowostawski & Poli [Now99] emerge como una alternativa más comprehensiva y específica. A su vez, se presenta como una clasificación más flexible y útil, al incluir categorías para modelos no tradicionales, no existentes en la taxonomía clásica, como los AG desordenados paralelos o los AGP de estado estacionario.

Es posible vincular la categorización de Nowostawski & Poli con los modelos desarrollados a lo largo de la historia de investigación de los AGP.

El *modelo maestro-esclavo* generalmente ha conservado esta denominación a lo largo de la evolución de las categorizaciones. De todos modos ciertos autores han designado con otros términos al modelo, a saber:

- han sido denominados *modelos de paralelismo global* por autores como Gordon & Whitley [GoW93], Schwehm [Sch96], y Tomassini [Tom95], [Tom99]
- como *modelo de paralelismo estándar* por Adamidis [Ada94a]
- como *modelo micrograno* por Lin et al. [Lin94].
- como *modelo de implementación centralizada* por Bianchini & Brown [BiB93] y por Baraglia & Perego [BaP99]

El *modelo distribuido de subpoblaciones con migración* es el que ha recibido mayor variedad de denominaciones por parte de los investigadores. Este modelo incluye a los siguientes enfoques:

- el *modelo PGA* de Petty & Leuze [Pet87]
- los *modelos de islas* de Gorges-Schleuter [Gor89], Gordon & Whitley [GoW93], Whitley [Whi90], [Whi97], [Whi98a], [Whi99] y Tomassini [Tom95], [Tom99]
- el *modelo forkingGA* de Tsutsui y Fujimoto [TsF93]
- el *modelo sin migración* de Shonkwiler [Sho93]
- el *modelo heterogéneo* de Tanese [Tan89b]
- el *modelo distribuido de grano fino* de Maruyama, Hirose & Konagaya [MHK93]
- el *modelo de implementación distribuida* de Bianchini & Brown [BiB93]
- el *modelo de descomposición* de Adamidis [Ada94a].
- el *modelo de AGP de grano grueso* de Lin et al. [Lin94] y de Baraglia & Perego [BaP99]
- el *modelo regional* de Schwehm [Sch96].
- el *modelo de AG distribuidos* de Alba y Troya [Alb99a]

El modelo denominado *de poblaciones estáticas solapadas sin migración* por Nowostawski corresponde al modelo de grano fino original de Robertson [Rob87] e incluye las siguientes propuestas:

- el *modelo de AGP de grano fino* de Mühlenbein [Müh89], Manderick & Spiessen [Man89] y Baraglia & Perego [BaP99].
- el *modelo de vecindades* de Gorges-Schleuter [Gor89],
- el *modelo de polinización* de Goldberg [Gol89a]
- el *modelo de AGP de grano fino* de Lin et al. [Lin94]
- el *modelo local* de Schwehm [Sch96].
- el *modelo estocástico* de Tomassini [Tom95], [Tom99]

La categoría de AG masivamente paralelos incluye a los modelos :

- modelo mpdGA de Baluja [Bal92], [Bal93a]
- el modelo de AGP celulares de Gordon & Whitley [GoW93], Tomassini [Tom98] y Alba y Troya [Alb99a]

El modelo de poblaciones dinámicas, con solapamiento es reciente y original de Nowostawski & Poli [KwN97], [Now98a], [Now98b], [NoP99a]. por lo cual no existen referencias directas sobre otros modelos similares.

La categoría de AGP de estado estacionario incluye las siguientes propuestas:

- el *modelo de islas* de Levine [Lev93], [Lev94],
- el *modelo maestro-esclavo* de Davidson y Rasheed [DaR99]
- el *modelo híbrido multipoblación (PHGA)* de Berger, Barkaoui & Bräysy [BBB01]

Los esfuerzos por diseñar AG desordenados paralelos han sido limitados, en esta línea de trabajo pueden mencionarse :

- el modelo de distribución de la evaluación de la función de fitness propuesto por Goldberg [Gol90]
- el modelo de distribución de datos en la fase primordial Merke & Lamont [MeL93]
- el modelo fmGA (*fast messy genetic Algorithm*) propuesto por Lamont, Gates & Merkle [LGM97]

Los modelos híbridos engloban a aquellos que combinan enfoques y estrategias, conjuntamente con los que cuentan con originales propuestas que aún no han constituido un tópico de investigación por sí mismos. Dentro de ellos es posible incluir a :

- el *modelo de comunidad* de Goldberg [Gol89a]
- los *modelos de AG cooperativos* de Adamidis [AdP96], [Ada97] y Venkateswaran, Obradovic & Raghavendra [VOR96]
- el modelo distribuido jerárquico con meta-optimización de parámetros de Masakazu et al. [Mas93]
- el modelo de organización en clusters denominado *modelo de implementación semidistribuido* de Bianchini & Brown [BiB93]
- el modelo distribuido con secciones compartidas de Hiroyasu, Miki & Watanabe [HMW99].
- los denominados *modelos jerárquicos* por Cantú-Paz [CaP00c]

De acuerdo a la enumeración precedente, es claro que la propuesta de unificación de terminologías y taxonomías de modelos de algoritmos genéticos paralelos constituye una necesidad importante para la comunidad científica interesada en este tema de investigación.

La alternativa de enfoques es variada, y determinar el modelo adecuado para la resolución de un determinado problema de búsqueda u optimización no será tarea sencilla sin tener un panorama de las características de los modelos, sus particularidades de implementación y las ventajas y desventajas respecto a los restantes enfoques.

En tal sentido, la intención de este trabajo es presentar las propuestas realizadas a lo largo del período de investigación en el área de AGP, resumiendo los detalles principales de los artículos, textos y reseñas disponibles.

Complementando el trabajo de categorización de Nowostawski & Poli, este capítulo reúne los modelos más relevantes de AGP y las enmarca dentro de la propuesta de taxonomía unificada.

Glosario

A continuación se describen de modo general abreviaturas utilizadas a lo largo del artículo, así como términos propios del vocabulario de la computación evolutiva que pueden resultar extraños al lector.

AG : Algoritmo(s) Genético(s)

AGP : Algoritmo(s) Genético(s) Paralelo(s)

Deme : término utilizado para designar una subpoblación que evoluciona semi-independientemente de otras poblaciones en un modelo multipoblacional.

0-1 ILP : problema de Programación Lineal Entera Booleano, las variables sólo pueden tomar los valores 0 ó 1.

Población panmíctica : población en la cual todos y cada uno de los individuos tiene una probabilidad no nula de interactuar con cualquier otro.

TSP : Traveling Salesman Problem, problema de optimización clásico que consiste en hallar un grafo de cubrimiento de costo mínimo.

Función de fitness : función que permite evaluar el grado de adaptación de un individuo, tomando en cuenta que tan buena solución es al problema de optimización que se intenta resolver.

Referencias bibliográficas

- [AbA92] A Parallel Genetic Algorithm for Solving the School Timetabling Problem
D. Abramson, J. Abela
Proceedings of the 15th Australian Computer Science Conference, 1992 ,
Vol. 14 pp. 1-11
- [ACT99a] Numerical and Real Time Analysis of Parallel Distributed GAs with
Structured and Panmictic Populations.
Enrique Alba, Carlos Cotta, Jose M Troya.
Proceedings of the IEEE Conference on Evolutionary Computing (CEC),
Volume 2, IEEE Press, pp. 1019-1026, 1999
- [ACT99b] Entropic and Real-Time Analysis of the Search with Panmictic, Structured,
and Parallel Distributed Genetic Algorithms.
Enrique Alba, Carlos Cotta, Jose M Troya.
Proceedings of the GECCO, W. Banzhaf et al. (eds.), Morgan Kaufmann,
pp. 773, 1999.
LCC Technical Report ITI 99-7, 1999.
- [Ada94a] Parallel Evolutionary Algorithms: A Review,
P. Adamidis
HERCMA'98, 4th Hellenic-European Conference on Computer Mathematics
and its Applications, Atenas, Grecia, 1998.
<http://aetos.it.teithe.gr/~adamidis/Papers/HERCMA98.ps.gz>
- [Ada94b] Review of Parallel Genetic Algorithms Bibliography
P. Adamidis
Technical Report, 1994, Automation & Robotics Lab., Dept. of Electrical &
Computer Eng., Aristotle Univ. of Thessaloniki, Grecia.
- [Ada97] Improving the performance of Parallel Genetic Algorithms with Co-
operating Populations with Different Evolution behavior
P. Adamidis
Ph.D. Thesis, Dept. of Electrical and Computer Eng., Aristotle Univ. of
Thessaloniki, Grecia, 1997
- [AdP96] Co-operating Populations with Different Evolution Behavior
Proceedings of 1996 IEEE International Conference on Evolutionary
Computation (ICEC'96), p. 188-191, Nagoya, Japani, 1996.
P. Adamidis, V. Petridis
- [AKP98] Advanced methods for evolutionary optimisation,
P. Adamidis, S. Kazarlis, V. Petridis
LSS'98, 8th IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Systems:
Theory and Applications, University of Patras, Grecia, 1998
- [AIT99a] A Survey of Distributed Genetic Algorithms.
Enrique Alba, Jose M Troya.
Complexity 4(4), pp.31-52, 1999
- [AIT99b] An Analysis of Synchronous and Asynchronous Parallel Distributed Genetic
Algorithms with Structured and Panmictic Islands
E. Alba, J. M. Troya.
Proceedings of the 11th IPPS/SPDP99 Workshops Held in Conjunction with
the 13rd International Parallel Processing Symposium and 10th Symposium
on Parallel and Distributed Processing
- [AIT01] Analyzing Synchronous and Asynchronous Parallel Distributed Genetic
Algorithms
Enrique Alba, Jose M Troya.
Future Generation Computer Systems, 17(4) pp. 451-465, Enero 2001.

- [AMP93] Parallelisation of a Genetic Algorithm for the Computation of Efficient Train Schedules
D. Abramson, G. Mills, S. Perkins
Proceedings of 1993 Parallel Computing and Transputers Conference, Brisbane, Nov. 1993, pp 139 - 149, IOS Press.

- [Bal93a] Structure and Performance of Fine Grain Parallelism in Genetic Search.
S. Baluja.
Proceedings of the Fifth International Conference (ICGA93), Morgan Kaufmann Publishers, San Mateo, CA., pp. 155-162, 1993.

- [Bal93b] A Parallel Genetic Algorithms, Premature Convergence and the nCUBE
Akarthik Balakrishnan
AComs 625x Term Project, Spring 1993

- [BBB01] A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows
Jean Berger, Mohamed Barkaoui, Olli Bräysy
Working paper, Defense Research Establishment Valcartier, Canada, 2001.

- [Bet76] Comparison of genetic algorithms and gradient based optimizers on parallel processors : Efficiency of use of precessing capacity
A. D. Bethke
Technical Report No 197, Ann Arbor, University of Michigan.
Citado en [CaP00c]

- [Bia95] Combining Distributed Populations and Periodic Centralized Selections in Coarse-Grain Parallel Genetic Algorithms
R. Bianchini, C. M. Brown, M. Cierniak, & W. Meira, Jr.
Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms 95, April 1995.

- [BiB93] Parallel Genetic Algorithms on Distributed-Memory Architectures
R. Bianchini, C. M. Brown.
Technical Report 436, University of Rochester. Computer Science Department, 1993.

- [Bos67] Mathematical optimization: are there abstract limits on natural selection?
W. Bossert
Mathematical Challenges to the Neo-Darwinian intepretation of evolution, Wistar Inst. Press, 5, pp. 35-46, 1967.
Citado en [CaP00c]

- [Bra90] On solving travelling salesman problems by genetic algorithms
H. Braun
Parallel Problem Solving from Nature, Proceedings of 1st Workshop, 1990
Volume 496 of Lecture Notes in Computer Science, pp. 129--133. Springer-Verlag, 1991.

- [BaP99] Parallel Genetic Algorithms for Hypercube Machine
R. Baraglia, R. Perego
Proceedings of Third International Conference for Vector and Parallel Processing - VECPAR 98, Porto, Portugal, June, 1998, 21-23, 1998, pp. 749-761.
Lecture Notes In Computer Science 1573, Springer, pp. 691-703, 1999.

- [CaP98a] A Summary of Research in Parallel Genetic Algorithm
E. Cantu-Paz
Illinois Genetic Algorithms Laboratory, Enero 1998.

- [CaP98b] Designing efficient master-slave parallel genetic algorithms.
E. Cantú-Paz.
Genetic Programming: Proceedings of the Third Annual Conference. (pp. 455). San Francisco, CA: Morgan Kaufmann, 1998.
- [CaP99] On the scalability of parallel genetic algorithms.
E. Cantú-Paz, D. Goldberg
Evolutionary Computation. 7(4), 429-449, 1999.
- [CaP00a] Markov chain models of parallel genetic algorithms.
E. Cantú-Paz
IEEE Transactions on Evolutionary Computation.4(3), 216-226, 2000
- [CaP00b] Selection intensity in genetic algorithms with generation gaps.
E. Cantú-Paz
GECCO-2000: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 911-918, San Francisco, CA, Morgan Kaufmann, 2000
- [CaP00c] Efficient and Accurate Parallel Genetic Algorithms
E. Cantú-Paz
Kluwer Academic Publishers, 2001
- [CaP01] Migration policies, selection pressure, and parallel evolutionary algorithms.
E. Cantú-Paz
Journal of Heuristics. 7(4), 311-334, 2001.
- [Coh87] J. P. Cohoon, S. U. Hedge, W. N. Martin, and D. S. Richards
Punctuated Equilibria: A Parallel Genetic Algorithm,
Proceedings of the Second International Conference on Genetic Algorithms,
pp. 148-154, 1987.
- [DaR99] Effect of Global Parallelism on the Behavior of a Steady State Genetic Algorithm for Design Optimization
B. Davison, K. Rasheed
Proceedings of the Congress of Evolutionary Computation CEC99 p. 534-541, 1999.
- [Dav91] Handbook of Genetic Algorithms
L. Davis.
Van Nostrand Reinhold, New York, 1991.
- [DeJ75] K. De Jong.
An analysis of the behaviour of a class of genetic adaptive systems.
PhD thesis, University of Michigan.
Citado en [CaP00c]
- [Den84] Referencia de Francis Crick a la denominada ley de Orgel, citada en el texto de Daniel C. Dennett, Elbow Room : the varieties of free will worth wanting (The MIT Press 1984)
- [Dor93] Implicit Parallelism in Genetic Algorithms
A. Bertoni, M. Dorigo
Artificial Intelligence (61) 2, 307-314, 1993.
- [Fog90] Implementing the Genetic Algorithm on Transputer Based Parallel Processing Systems
Terence Fogarty, Runhe Huang
Parallel Problem Solving from Nature 1990, pp. 145-149.
- [Gol89a] Genetic Algorithms in Search, Optimization and Machine Learning.
G. E. Goldberg
Addison Wesley, New York, 1989.

- [Gol89b] Sizing populations for serial and parallel genetic algorithms.
D. E. Goldberg
Proceedings of the Third International Conference on Genetic Algorithms,
70-79, 1989.
Disponible como TCGA Report 88004.

- [Gol90] Messy genetic algorithms: Motivation, analysis, and first results.
D. E. Goldberg, B. Korb, and K. Deb.
Complex Systems, 3:493--530, 1990.

- [Gor89] ASPARAGOS: An asynchronous parallel genetic optimization strategy.
M. Gorges-Schleuter
Proceedings of the Third International Conference on Genetic Algorithms,
pp.422-427, San Mateo CA, 1989, Morgan Kaufmann.

- [Gor90] M. Gorges-Schleuter. Explicit parallelism of genetic algorithms through
population structures.
Proceedings of 1st PPSN'90, pages 150-159, 1990.

- [Gor92] Comparison of Local Mating Strategies in Massively Parallel Genetic
Algorithms.
Martina Gorges-Schleuter
Proceedings of 2nd PPSN 1992, 559-568.

- [Gol93] Rapid, accurate optimization of difficult problems using fast messy genetic
algorithms.
D. Goldberg, K. Deb, H. Kargupta, & G. Harik.
Proceedings of the Fifth International Conference on Genetic Algorithms,
pp. 56-64. Morgan Kaufman CA, 1993.
Disponible como IlliGAL Report 93004.

- [Gor97] "Asparagos96 and the traveling salesman problem",
M. Gorges-Schleuter
Proceedings of 1997 IEEE International Conference on Evolutionary
Computation, pp. 171-174. IEEE, April 1997

- [Gor98] A Comparative Study of Global and Local Selection in Evolution Strategies
Martina Gorges-Schleuter
Parallel Problem Solving from Nature {PPSN V} pp. 367-377, 1998.

- [GoW93] Serial and Parallel Genetic Algorithms as Function Optimizers
V. Gordon, D. Whitley
Proceedings of the Fifth International Conference on Genetic Algorithms,
Kauffman CA, pp. 177-183, 1993.

- [Gre81] Parallel adaptive algorithms for function optimization.
John J. Grefenstette
Technical Report CS-81-19, Computer Science Department, Vanderbilt
University, Nashville, TN, 1981
Citado en [Gol89a].

- [Gre89] How genetic algorithms work: A critical look at implicit parallelism.
J. Grefenstette y J. Baker
Proceedings of the Third International Conference on Genetic Algorithms,
San Mateo 1989, Morgan Kaufmann, 20-27.

- [Gre91] Conditions for implicit parallelism.
J. Grefenstette
Foundations of Genetic Algorithms, G.J.E. Rawlins (Ed.), San Mateo, CA:
Morgan Kaufmann.

- [Gre95] Robot Learning With Parallel Genetic Algorithms Onnetworked Computers
John J. Grefenstette
Proceedings of the Summer Computer Simulation Conference (SCSC '95).
Society for Computer Simulation, Ottawa, Ontario, Canada, 1995.
- [Gro85] Computer simulations of genetic adaptation: Parallel Subcomponent
Interaction in a Multilocus Model
P. Grosso
PhD thesis, University of Michigan, 1985.
Citado en [CaP00c].
- [Har96] Analysis of the Numerical Effects of Parallelism on a Parallel Genetic
Algorithm
W. Hart, S. Baden , R. Belew, S. Kohn
Proceedings of the 10th International Parallel Processing Symposium, pp.
606-612. 1996
- [HMW99] T. Hiroyasu, M. Miki, S. Watanabe
Distributed Genetic Algorithms with a New Sharing Approach in
Multiobjective Optimization Problems.
IEEE Proceedings of the 1999 Congress on Evolutionary Computation,
pp.69-76, 1999.
- [Hol59] A universal computer capable of executing an arbitrary number of
subprograms simultaneously.
J. H. Holland
Proceedings of the Eastern Joint Computer Conference, 1959, pp. 108-112.
Citado en [Gol89a]
- [Hol60] Iterative circuit computers
J. H. Holland
Proceedings of the 1960 Western Joint Computer Conference, 1960, pp.
259-265.
Citado en [Gol89a]
- [Hol63] Outline for a logical theory of adaptive systems
J. H. Holland
Journal of the Association for Computer Machinery, 1963, 3, pp.297-314
Citado en [Gol89a]
- [Hol75] Adaption in Natural and Artificial Systems
J. H. Holland
Univerity of Michigan Press, Ann Arbor, 1975.
Citado en [Gol89a]
- [KwN97] The Search Engine for Information Systems Based on Parallel Genetic
Algorithm
H. Kwasnicka, M. Nowostawski
Proceedings of the 2nd International Conference on Parallel Processibg &
Applied Mathematics, pp. 442-451, Czestochowa 1997.
- [Lev93] A Genetic Algorithm For The Set Partitioning Problem
D. Levine
Proceedings of the Fifth International Conference on Genetic Algorithms,
San Mateo, Morgan Kaufmann, 1993, pp. 81--487
- [Lev94] A Parallel Genetic Algorithm for the Set Partitioning Problem.
D. Levine.
PhD Thesis, Illinois Institute of Technology, 1994.

- [LGM97] A MPI Implementation of the Fast Messy Genetic Algorithm
Gary Lamont, George Gates & Laurence Merkle
Proceedings of the Intel Supercomputer Users Group Conference (ISUG'97), 1997.
- [LGM98] Scalability of a MPI-Based Fast Messy Genetic Algorithm
Gary Lamont, George Gates & Laurence Merkle
Proceedings of ACM Symposium on Applied Computing (SAC'98), 1998.
- [Lin94] Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach
Lin S.C., Punch III W.F., Goodman E.D.
Proceedings of the Sixth IEEE Parallel & Distributed Processing, IEEE Press, pp. 28-37, 1994
- [Man89] Fine-Grained Parallel Genetic Algorithms
Bernard Manderick, Piet Spiessens
Proceedings of the Third International Conference on Genetic Algorithms, pp. 428-433, 1989.
- [Mas93] A Cooperative Search Strategy Using Hierarchical Genetic Algorithms and Its Implementation on the UNIX-Network
Takahashi Masakazu, Munetomo Masaharu, Takai Yoshiaki, Sato Yoshiharu
IPSJ SIGNotes Contents Artificial Intelligence No.091
- [MeL93] Comparison of Parallel Messy Genetic Algorithms Data Distribution Strategies
Laurence Merkle, Gary Lamont
Proceedings of Fifth International Conference on Genetic Algorithms, pp. 191-198, 1993.
- [MHK93] A Fine-Grained Parallel Genetic Algorithm for Distributed Parallel Systems
T. Maruyama, T. Hirose, A. Konagaya.
Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann CA, pp 184-190, 1993.
- [MTS93] An Efficient Migration Scheme for Subpopulation-Based Asynchronously Parallel Genetic Algorithms
M. Munetomo, Y. Takai, Y. Sato
Proceedings of the Fifth International Conference on Genetic Algorithms, Kauffman CA, 1993, pp. 649.
- [Müh88] Evolution algorithms in combinatorial optimization.
H. Mühlenbein, M. Gorges-Schleuter, and O. Kramer.
Parallel Computations, 7, pp. 65-85, 1988.
- [Müh89] H. Muhlenbein.
Parallel genetic algorithms, population genetic and combinatorial optimization.
Proceedings of the Third International Conference on Genetic Algorithms, pp 416-421, Arlington, 1989.
- [Müh91] Evolution in Time and Space - The Parallel Genetic Algorithm
Heinz Mühlenbein
Foundations of Genetic Algorithms, Morgan-Kaufman, pp. 316-337, 1991.
- [NoP99a] Dynamic Demes Parallel Genetic Algorithm
Mariusz Nowostawski, Riccardo Poli
Proceedings of Third International Conference on Knowledge-based Intelligent Information Engineering Systems KES'99 (Adelaide, South Australia, 31 August - 1 September 1999).

- [NoP99b] Parallel Genetic Algorithms Taxonomy
Mariusz Nowostawski, Riccardo Poli
Proceedings of Third International Conference on Knowledge-based Intelligent Information Engineering Systems KES'99 (Adelaide, South Australia, 31 August - 1 September 1999).
- [Now98a] Parallel Genetic Algorithms in Geometry Atomic Cluster Optimization and Others Applications
Mariusz Nowostawski
M.S. thesis, School of Computer Science, The university of Birmingham, UK, Setiembre 1998.
- [Now98b] Parallel Genetic Algorithms and Sequencing Optimisation
Mariusz Nowostawski
Mini-project report, 1998.
Online (<http://marni.otago.ac.nz/~mariusz/gzipped/rep2r.ps.gz>)
consultado junio 2002
- [Pet87] Genetic Algorithms and their implementation
Chrisila B. Pettey, Michael R. Leuze, John J. Grefenstette
Proceedings of the Second International Conference on Genetic Algorithms, Morgan Kauffmann Publishers, pp. 155-161, 1987.
- [Pet89] A Theoretical Investigation of a Parallel Genetic Algorithm
Chrisila B. Pettey, Michael R. Leuze
Proceedings of the Third International Conference on Genetic Algorithms, 1989, pp. 398-405I, 1989.
- [Pun93] Further Research on Feature Selection and Classification using Genetic Algorithms
W. Punch, E. Goodman, M. Pei, L. Chai-Shun, P. Hovland, R. Enbody
Proceedings Fifth ICGA, Junio 1993, pp. 557-564
- [Ras98] GADO: A genetic algorithm for continuous design optimization
Khaled Rasheed
Technical Report DCS-TR-352. Department of Computer Science, Rutgers University. New Brunswick, NJ. Ph.D., 1998.
- [Rob87] G. Robertson
Parallel Implementation of Genetic Algorithms in a Classifier System
Proceedings of the Second International Conference on Genetic Algorithms, pp. 140-147, 1987.
- [Sch96] Parallel Population Models for Genetic Algorithms
Markus Schwehm
Foundations of Genetic Algorithms Workshop (FOGA4), San Diego, California, 1996.
- [Sho93] Parallel Genetic Algorithms
R. Shonkwiler.
Proceedings of the Fifth International Conference on Genetic Algorithms, Kauffman CA, 1993, pp. 658.
- [Spr99] A Unified Model of Non-Panmictic Population Structures in Evolutionary Algorithms.
J. Sprave.
Technical Report ISSN 1433-3325 Universität Dortmund, 1999.
- [Sta91] Optimization Using Distributed Genetic Algorithms
T. Starkweather, D. Whitley, K. Mathias
Parallel Problem Solving from Nature, p. 176-185, Springer-Verlag (Berlin), 1991.

- [Tan89a] Distributed Genetic Algorithms for Function Optimization
Reiko Tanese
Ph. D. Thesis, Department of Electrical Engineering and Computer Science,
University of Michigan, 1989.
- [Tan89b] Distributed Genetic Algorithms
Reiko Tanese
Proceedings of the Third International Conference on Genetic Algorithms,
Kauffman, San Mateo, 1989, pp 434-439.
- [TFG97] Forking GAs: GAs with Search Space Division Schemes
S. Tsutsui, Y. Fujimoto, A. Ghosh
Evolutionary Computation, MIT Press, Vol. 5, No. 1, pp. 61-80, 1997.
- [Tom93a] Massively parallel evolutionary algorithms
M. Tomassini
Proceedings of the 2nd Connection Machine User Meeting, 1993.
- [Tom93b] The Parallel Genetic Cellular Automata: Application to Global Function
Optimization
M. Tomassini
Artificial Neural Nets and Genetic Algorithmspp. 385-391, 1993.
- [Tom95] A Survey of Genetic Algorithms.
M Tomassini.
Annual Reviews of Computational Physics, volume III, pp. 87-118, 1995.
Disponible como Technical Report 95/137, Department of Computer
Science, Swiss Federal Institute of Technology, Lausanne, Switzerland,
Julio, 1995.
- [Tom99] Parallel and Distributed Evolutionary Algorithms: A Review
M. Tomassini
Evolutionary Algorithms in Engineering and Computer Science, pp. 113-
133, 1999.
- [TsF93] Forking Genetic Algorithm with Blocking and Shrinking Modes (fGA)
S. Tsutsui, Y. Fujimoto
Proceedings of the 5th International Conference on Genetic Algorithm
(ICGA), pp. 206-213, 1993.
- [TsF94] Extended Forking Genetic Algorithm for Order Representation (o-fGA)
S. Tsutsui, Y. Fujimoto
Proceedings of the First IEEE International Conference on Evolutionary
Computation (ICEC'94) , Vol. 1, pp. 170-175, 1994.
- [VOR96] Cooperative Genetic Algorithm for Optimization Problems in Distributed
Computer Systems
R. Venkateswaran, Zoran Obradovi'c, C.S.Raghavendra
Proceedings of the Second Online Workshop on Evolutionary Computation,
pp. 49-52, 1996
- [Whi89] The GENITOR Algorithm and Selection Pressure : Why Rank-Based
Allocation of Reproductive Trials is Best.
D. Whitley.
Proceedings of the Third International Conference on Genetic Algorithms,
Kauffman, San Mateo, 1989, pp 116-121.
- [Whi90] GENITOR II: a Distributed Genetic Algorithm.
D. Whitley, T. Starkweather.
Journal of Experimental and Theoretical Artificial Intelligence 2, pp 189-
214, 1990.

- [Whi93] Cellular Genetic Algorithms
D. Whitley
Proceedings of the Fifth International Conference on Genetic Algorithms.
Morgan Kaufmann, San Mateo, 1993, pp. 658.
- [Whi97] Messy Genetic Algorithms for Subset Feature Selection.
D. Whitley, R. Beveridge, C. Guerra and C. Graves.
International Conference on Genetic Algorithms. T. Baeck, ed. Morgan Kaufmann, 1997.
- [Whi97] Island Model genetic Algorithms and Linearly Separable Problems
Darrell Whitley, Soraya B. Rana, Robert B. Heckendorn
Evolutionary Computing, AISB Workshop 1997, pp. 109-125
- [Whi99] The Island Model Genetic Algorithm: On Separability, Population Size and Convergence.
Darrel Whitley, Soraya Rana, Robert B Heckendorn.
Special Issue on Evolutionary Computing Volume 7, Number 1, 1999, pp. 33-47
- [Whi98a] Representation Issues in Neighborhood Search and Evolutionary Algorithms
D. Whitley, S. Rana, R. Heckendorn
Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Pp. 39-57, John Wiley, 1998.
- [Whi98b] GENITOR: A Different Genetic Algorithm.
D. Whitley, J. Kauth.
Technical Report CS-88-101, Colorado State University, Department of Computer Science, Fort Collins, 1998.