

# **Evolución en el diseño y clasificación de Algoritmos Genéticos Paralelos**

**Sergio Nesmachnow**

Universidad de la República, Instituto de Computación  
Montevideo, Uruguay, 11300  
sergion@fing.edu.uy

## **Abstract**

This paper briefly describes the application of high performance techniques (parallel and distributed processing) proposed by researchers in the field of evolutionary computing, dedicated to improve the performance and the search quality of genetic algorithms. This work reviews the state of art on parallel genetic algorithms, showing the evolution on design and classification of parallel techniques applied to genetic algorithms over the last twenty-five years.

An extended version of this paper, including a more detailed description of parallel processing techniques applied to genetic algorithms and its classifications can be found on the Technical Report [57].

**Keywords:** Genetic Algorithms, parallel and distributed systems.

## **Resumen**

El objetivo de este trabajo consiste en brindar una descripción de los mecanismos de aplicación de las técnicas de procesamiento de alta performance (procesamiento paralelo y distribuido) propuestos 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. El trabajo conforma una reseña del estado del arte orientada a exponer la evolución en el diseño y la clasificación de los enfoques de paralelismo aplicados a los algoritmos genéticos en los últimos veinticinco años.

Una versión extendida de este trabajo, que ofrece una descripción más detallada de los modelos de paralelismo aplicados a los algoritmos genéticos y sus clasificaciones puede encontrarse en el Reporte Técnico [57]

**Palabras claves:** Algoritmos genéticos, sistemas paralelos y distribuidos.

## 1. Introducción

Los algoritmos evolutivos basan su funcionamiento en un mecanismo análogo a los procesos evolutivos naturales, con el objetivo de resolver problemas de búsqueda y optimización. En el caso de los Algoritmos Genéticos (AG), durante el proceso 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 una *función de fitness*. Una descripción detallada del mecanismo de los AG puede encontrarse en el texto de Goldberg [30].

Las técnicas de procesamiento paralelo y distribuido se aplican al modelo clásico de AG con el objetivo de obtener mejoras desde el punto de vista de la eficiencia y para perfeccionar la calidad de la búsqueda genética. Desde la perspectiva de la eficiencia, paralelizar un AG permite afrontar la lentitud de convergencia para problemas cuya dimensión motiva el uso de poblaciones numerosas, o múltiples evaluaciones de funciones de fitness costosas. Desde el punto de vista algorítmico, los Algoritmos Genéticos Paralelos (AGP) pueden explotar el paralelismo intrínseco del mecanismo evolutivo, trabajando simultáneamente sobre varias poblaciones semi-independientes para resolver el mismo problema. Eventuales intercambios de soluciones (*migraciones*) introducen diversidad para evitar problemas de convergencia en óptimos locales. Complementariamente, los AGP pueden aprovechar características de paralelismo propias del problema, analizando concurrentemente diferentes secciones del espacio de búsqueda.

El modelo que surge de distribuir funcionalmente el algoritmo, asignando a diferentes procesadores distintas etapas del mecanismo evolutivo, se conoce como modelo maestro-esclavo. Es usual distribuir la evaluación de la función de fitness, que generalmente involucra un tiempo de ejecución mayor que el de los sencillos operadores evolutivos. Por otra parte, el enfoque orientado a la distribución de datos da lugar a un modelo de población distribuida, organizada en subpoblaciones semi-independientes. Estos dos enfoques, reconocidos desde las primeras etapas de la investigación en el área, han evolucionado y se han diversificado, dando lugar a múltiples modelos de AGP. Excepto para el modelo maestro-esclavo sincrónico, los AGP no tienen el mismo comportamiento que el algoritmo serial. Al utilizar poblaciones semi-independientes, la interacción panmíctica (donde cada individuo tiene la posibilidad de interactuar –competir o cruzarse– con cualquier otro) se reemplaza por un mecanismo de interacción local, ocasionando que el algoritmo distribuido sea diferente. Esta característica proporciona nuevos matices de interés sobre el problema paralelizado, dependiendo de los diferentes criterios utilizados al dividir el problema.

Analizando las ideas propuestas, es posible identificar los criterios utilizados para clasificar los AGP:

- El modo de evaluación de la función de fitness, que puede ser concentrada o distribuirse en diferentes procesos.
- El número de poblaciones del AGP, puede utilizarse una única población (panmixia), o 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.
- Los individuos sobre los cuales se aplica el mecanismo de selección, pueden ser un conjunto local o uno global.
- El mecanismo de sincronización entre los elementos de procesamiento.

El modo de organizar la población es el principal criterio de clasificación utilizado por los investigadores en el área.

En los últimos veinticinco años, una gran cantidad de propuestas de implementación y modelos teóricos han sido presentados para los AGP. Analizando su contenido, es posible distinguir tres etapas en la investigación, relevantes para comprender los conceptos teóricos, de diseño e implementación involucrados.

La primera etapa corresponde a los orígenes del modelo y se extiende hasta el fin de la década de 1980. En esta etapa se realizaron análisis teóricos, propuestas de diseño y las primeras implementaciones de AGP.

La etapa siguiente se caracteriza por una prolífica gama de AGP implementados sobre plataformas específicas de hardware de alto desempeño. La etapa comprende la primera mitad de la década de 1990, claramente influida por la difusión de las técnicas de procesamiento paralelo, en particular sobre multicomputadores de memoria compartida.

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 formalizar los conceptos teóricos y el modelado matemático del mecanismo de los AGP como herramientas 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 las secciones siguientes se analizan las etapas, aludiendo a trabajos considerados representativos de cada período.

## 2. Las Investigaciones Pioneras

La naturaleza paralela de los algoritmos evolutivos fue reconocida en los trabajos pioneros en el área en la década de 1960. Holland [42] y Bossert [18] no propusieron implementaciones de AGP, pero sus análisis de las propiedades de los modelos seriales facilitaron la comprensión del mecanismo evolutivo como herramienta de optimización.

Deben recorrerse más de diez años para encontrar una nueva referencia a la conjunción de la programación evolutiva y las técnicas de paralelismo. En 1975, De Jong [27] analizó teóricamente el modelo maestro-esclavo sincrónico, y propuso definir un valor de salto  $G \in [1/n, 1]$  para la fracción de la población reemplazada en cada generación.

En 1976, Bethke [15] realizó el estudio teórico de un AG sobre una máquina paralela SIMD, analizando la eficiencia de uso de la capacidad de procesamiento. Concluyó que el máximo de eficiencia se obtiene cuando se evalúan funciones de fitness mucho más costosas que las operaciones de evolución, un caso típico para muchas aplicaciones.

La primera referencia a una taxonomía de los AGP corresponde a Grefenstette en 1981 [40]. En su clasificación reconoce tres categorías, el modelo maestro-esclavo basado en la distribución de la función de fitness (del cual describe la versión sincrónica y la asincrónica), un modelo para máquinas paralelas de memoria compartida y un modelo distribuido de poblaciones múltiples. A grandes rasgos, estas categorías han evolucionado en el tiempo, refinándose para aplicarse a diseños específicos y/o generalizándose al plantear taxonomías de amplio espectro.

En 1987 Pettey et al. [62] presentan la primera implementación de un AGP distribuido, bajo el genérico nombre de *Parallel Genetic Algorithm* (PGA). Este algoritmo utiliza un modelo de subpoblaciones homogéneas cooperativas (“islas”) con topología de migración dinámica. El estudio preliminar del PGA fue empírico, analizándose teóricamente en los años siguientes por Pettey y Leuze [63] y Mühlenbein [54].

Trabajando sobre variantes del modelo PGA, Tanese [70] analizó los efectos de la migración en la eficiencia y calidad de resultados de los AGP. Mostró que tasas medianas de migración son necesarias para hallar soluciones de la misma calidad que un AG serial y concluyó que pueden obtenerse speedups casi lineales cuando la migración es infrecuente y de tasa baja. Complementariamente, estudió la influencia de los parámetros de migración en la pérdida de diversidad de soluciones [71]. Para prevenirla, propuso fijar diferentes parámetros y/o mecanismos de selección y cruzamiento para cada isla, en una de las primeras referencias de un AGP heterogéneo encontradas en la literatura.

Robertson introdujo los AGP *de grano fino* en 1987 [65], implementados sobre un equipo CM1, paralelizando todas las fases del algoritmo (selección de progenitores, selección de población reemplazada, cruzamiento y mutación). Su modelo reportó buenos resultados, con un tiempo de ejecución independiente del tamaño de la población.

Simultáneamente, los AGP *de grano fino* fueron estudiados por Manderick y Spiessen [49], Mühlenbein [52] [53] y Gorges-Schleuter [35] proponiendo fijar los individuos en una grilla, restringiendo la selección y el cruzamiento a pequeñas vecindades en la grilla. ASPARAGOS, el modelo de Mühlenbein y Gorges-Schleuter, fue implementado sobre una red de transputers, con la población estructurada en una escalera cíclica. Luego evolucionó incluyendo nuevas estructuras y mecanismos de pareo [38] hasta constituirse en una herramienta efectiva de optimización [39].

En su texto de 1989 [30], Goldberg describe dos modelos para los AGP: el modelo de *comunidad* (un híbrido con migración entre poblaciones, cada una de las cuales 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.

En 1990, Gorges-Schleuter [36] ofrece una clasificación de los modelos que utilizan poblaciones no panmíticas:

- El *modelo de islas*, con poblaciones separadas y migración, siendo cada isla un AG con población global.
- El *modelo stepping stone*, que cuenta con islas organizadas 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, refiriéndose a ambos como *modelo de islas*.

### 3. La Influencia del Desarrollo de las Arquitecturas Paralelas

La proliferación de arquitecturas de alta performance al inicio de la década de 1990 influyó positivamente en el desarrollo de versiones paralelas de AG. La investigación se movió desde el terreno teórico al ámbito de las implementaciones y su aplicación a la resolución de complejos problemas de optimización.

Simultáneamente, el análisis comparativo de estrategias de paralelismo surgió como un importante tema de estudio.

En esta etapa, que se extiende hasta mediados de la década de 1990, los investigadores centraron su trabajo en arquitecturas multiprocesador específicas, y los modelos de paralelismo aplicados trataron de obtener las mayores ventajas del hardware disponible. La tendencia a diseños específicos comienza a revertirse hacia el fin del lustro, surgiendo trabajos con el objetivo de condensar resultados y ofrecer modelos independientes de las arquitecturas.

En 1991, Starkweather et al. [69] investigaron bajo que condiciones un AGP proporcionaría una solución de mejor calidad que un AG serial. Especularon que los AGP trabajarían mejor para funciones de fitness separables, donde una buena solución es combinación de soluciones parciales, que en problemas con funciones no separables.

Fogarty y Huang [29] implementaron el modelo maestro-esclavo en una red de transputers, concluyendo que el crecimiento del overhead de comunicación constituía un impedimento para futuras mejoras de sus resultados. Este problema sería corroborado por varios investigadores en los años siguientes.

Abramson y Abela [1] implementaron un AGP panmítico sobre un equipo de memoria compartida (Encore Multimax de 16 procesadores) para resolver el problema de asignación de horarios, reportando speedups limitados.

En 1993, Abramson, et al. [2] agregaron una máquina de memoria distribuida (Fujitsu AP1000 de 128 procesadores) logrando buenos speedups al trabajar con hasta 16 procesadores, degradándose el speedup al aumentar su número.

En el trabajo de Gordon y Whitley [34] se propone una clasificación de AGP que reconoce tres categorías:

- AGP *de población global*, una versión del modelo maestro-esclavo que utiliza selección por torneo y sus versiones elitistas para simplificar el paralelismo.
- AGP con *modelo de islas* y migración, similar a los modelos de Gorges-Schleuter [36].
- AG *masivamente paralelos* o AG *celulares*, que asignan un número bajo de individuos por elemento de procesamiento (EP). 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. La denominación *celular* se justifica por comportarse el AGP como un tipo particular de autómatas celulares [82].

Los autores realizan un estudio comparativo de diferentes modelos de AGP, incluyendo a su propia implementación, denominada Genitor. Genitor nació en 1989 [80] como un AGP de población global y selección por rango, y evolucionó a un modelo de islas con migración en su versión Genitor II [81].

Un modelo de paralelismo trivial, denominado IIP (independent, identical & parallel) es propuesto por Shonkwiler, en 1993[67]. Asignando a varios procesadores AG seriales que resuelven una misma instancia del problema, permite obtener resultados estadísticos y trabajar con diferentes condiciones iniciales y combinaciones de los parámetros.

Baluja [11] modificó los AGP *masivamente paralelos*, introduciendo poblaciones estructuradas con solapamiento, que permiten la transferencia gradual de información genética sin la introducción súbita de cromosomas.

Para el diseño de su modelo *massively distributed parallel genetic algorithm* (mpdGA), Baluja analizó estructuras de las poblaciones y tamaños de las secciones solapadas, indicando que debe lograrse un compromiso entre la lentitud de propagación de soluciones para áreas de solapamiento pequeñas y la desviación de la idea de poblaciones múltiples al crecer las áreas y acercarse a un modelo panmítico. Su propuesta de implementación se basa en la arquitectura paralela MassPar MP-1, modelo SIMD, sobre la cual reporta obtener resultados de similar calidad, pero con mayor eficiencia que el modelo distribuido Genitor [81] para un conjunto estándar de problemas de prueba.

En 1993 Baluja [12] estudia distintas topologías para la estructura de población y su performance sobre un conjunto de problemas de prueba, sin llegar a conclusiones definitivas sobre la existencia de una configuración óptima.

El problema de la convergencia prematura es estudiado por Balakrishnan en [10], quien propone un modelo de migración sobre una arquitectura nCube para evitarlo. Los resultados obtenidos indican la eficiencia del mecanismo de paralelismo implementado para la impedir la convergencia anticipada aumentando la diversidad de la población.

El modelo *forkingGA* es propuesto por Tsutsui y Fujimoto [76], con la idea de utilizar poblaciones independientes para resolver problemas multimodales. El mecanismo de migración considera una población padre y varias hijas, responsables de diferentes sectores no solapados del espacio de búsqueda. El modelo fue extendido en [77] [78].

Uno de los primeros AGP de grano fino implementados sobre una red de workstations interconectadas por una LAN fue obra de Maruyama, Hirose y Konagaya [50]. En su trabajo se analizan los efectos del sincronismo entre poblaciones distribuidas sobre la calidad de la solución y la mejora de performance comparada con un AG serial.

En 1993, Bianchini y Brown [16] estudiaron el diseño de AGP sobre arquitecturas paralelas de memoria distribuida. Sobre estas plataformas, tradicionalmente se descartaban las implementaciones centralizadas, prefiriéndose modelos distribuidos con subpoblaciones de granularidad mayor, evitando los cuellos de botella en las comunicaciones.

Considerando ciertos casos donde la calidad de la búsqueda mejora al trabajar con una población global, los autores propusieron extender los modelos distribuidos de Grefenstette [40], introduciendo centralización en los algoritmos. El enfoque *semi-distribuido* propone resolver el equilibrio entre calidad de la búsqueda y performance del algoritmo, organizando los procesos en clusters sin generar un aumento excesivo en las comunicaciones.

De este modo, la propuesta comparte algunas de las ventajas de los modelos distribuidos y otras de los centralizados. Bianchini y Brown proponen una taxonomía de AGP, que toma en cuenta el modo de distribución de la población :

- Implementación *centralizada*: modelo maestro-esclavo de población única. Se introducen variantes para evitar los cuellos de botella en las comunicaciones, asignando varios individuos a los procesos esclavos y proporcionándoles la lógica para evolucionar aisladamente por un cierto número de generaciones.
- Implementación *semi-distribuida*: modelo que organiza clusters de procesadores, que trabajan al estilo del modelo centralizado, con esclavos dotados de lógica extendida y maestros de los clusters que intercambian sus mejores individuos. Esta organización reduce el problema de acceso a un único maestro, aunque incluye overhead para la replicación semi-distribuida y las nuevas comunicaciones entre procesos maestros.
- Implementación *distribuida*: modelo que elimina las poblaciones compartidas. Cada procesador mantiene una porción de la población y aplica un AG secuencial sobre ella, con migración de los mejores individuos.
- Implementación *totalmente distribuida*: corresponde al modelo anterior, sin migración entre subpoblaciones.

Los autores señalan que sus versiones *semi-distribuidas* sobre memoria distribuida mejoran la eficiencia de los modelos centralizados tradicionales sobre memoria compartida. Este resultado es confirmado cuantitativamente por

Bianchini et al. [17], testeando sobre funciones bien conocidas en benchmarks de optimización. En este trabajo se indica que las selecciones centralizadas eventuales introducen mejoras sobre los enfoques habituales de distribución de población, calificando la estrategia de centralizaciones como un mecanismo robusto, que permite lograr mejoras en la calidad de los resultados obtenidos, manteniendo los tiempos de ejecución y las comunicaciones bajo control.

Las técnicas de paralelismo también son aplicables al modelo de AG *desordenados* (Messy Genetic Algorithms original de Goldberg [32]). Un AG *desordenado* consta de una fase primordial en la cual se crea la población utilizando una estrategia de enumeración parcial, distribuyendo bloques de construcción de soluciones y una fase de yuxtaposición donde se mezclan las soluciones encontradas en la fase primordial. La fase primordial domina en cuanto al tiempo de ejecución y las técnicas de paralelismo se aplican para mejorar su performance.

Merkle y Lamont [51] proponen estrategias de distribución de datos en la selección por torneo de la fase primordial (indexado simple y modificado, ordenamiento e intercalado de bloques de construcción, ordenamiento y asignación por bloques). Conjuntamente, ofrecen una 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, prueban el efecto significativo de las estrategias de distribución de datos sobre el tiempo de ejecución de la fase primordial, dejando constancia de que la idea no mejora la calidad de la solución respecto al modelo secuencial.

En 1993, Munetomo et al [55] estudian los diferentes esquemas de migración que permiten reducir comunicaciones innecesarias en los modelos distribuidos. Proponen un esquema, denominado *algoritmo de intercambio sigma*, que toma en cuenta la desviación estándar en la distribución del fitness de cada deme para determinar la frecuencia de intercambio de material genético. En [56] definen un modelo de AGP con estructuras jerárquicas de subpoblaciones y mecanismo de optimización de los parámetros de cada AG utilizando un meta-AG de nivel superior. Este modelo se implementó sobre una red de workstations resultando un efectivo algoritmo de búsqueda distribuido.

Los artículos de Levine [46] [47] presentan una implementación paralela del modelo de AG de *estado estacionario*. Por oposición a los AG *generacionales*, que reemplazan una significativa parte de la población luego de los pareos, los AG de *estado estacionario* reemplazan una pequeña parte, 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 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, calidad de la solución hallada y capacidad de escalabilidad para resolver problemas más complejos.

Adamidis [3] [4] plantea una categorización de AGP basada en la operativa del mecanismo evolutivo. El *modelo estándar* corresponde a una paralelización global maestro-esclavo que implementa el AG serial, al no modificarse el algoritmo al distribuir la evaluación de la función de fitness. El *modelo de descomposición* tiene una operativa diferente a la del AG serial, ya que la población se encuentra distribuida. 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 más complejos, combinando los enfoques propuestos. En esta línea de investigación, el autor propuso un *modelo cooperativo* en su tesis de doctorado [5], que utiliza diferentes mecanismos de selección y operadores en las poblaciones.

Lin et al. [48] ofrecen una clasificación de AGP que describe las tres categorías clásicas, especializándose en los AGP de grano grueso. Las categorías de su clasificación corresponden a :

- AGP *micrograno* : modelo de población única, con distribución tipo maestro-esclavo de la evaluación de la función de fitness. Este modelo logra mejoras significativas de performance respecto al AG serial para problemas complejos, pero no resuelve el problema de convergencia prematura.
- AGP *de grano fino* : modelo que asigna un individuo a cada elemento de procesamiento (EP). Cada individuo forma parte de varias subpoblaciones, solapadas entre sí, que determinan la adyacencia entre individuos de acuerdo a una topología de conexión. La alta conectividad entre vecinos incrementa la difusión de individuos aptos, pero hace a la población susceptible al dominio de algún esquema de codificación que podría llevar a una convergencia prematura. Como al limitar las interacciones se reduce la performance, se plantea que el modelo debe lograr un compromiso entre estos factores para obtener soluciones de calidad, con buena performance.
- AGP *de grano grueso* : modelo con una gran cantidad de individuos asignados a poblaciones 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 el modelo aislado, donde las poblaciones evolucionan independientemente sin existir comunicación, y los modelos de migración, que puede ocurrir en un punto común para las subpoblaciones (modelo de islas sincrónico) o en eventos no relacionados con la evolución (modelo de islas asincrónico)
  - El esquema de conectividad entre poblaciones, que puede ser estático, basado en las topologías de los multicomputadores, 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 en sus subpoblaciones. Un AGP no homogéneo permite a

las subpoblaciones utilizar diferentes mecanismos evolutivos, con criterios propios para seleccionar parámetros y codificaciones para las soluciones. La variación de representaciones y esquemas evolutivos es beneficiosa para problemas de optimización multiobjetivo o con espacios de búsqueda extensos.

Lin et al. proponen un AGP heterogéneo, el *AG Injection Island* (AGII). El AGII utiliza diferentes resoluciones (largos de representación) para las codificaciones en cada población y su mecanismo de migración es de “una vía”, exclusivamente desde poblaciones de baja resolución (“padres”) a poblaciones de alta resolución (“hijos”).

Tomando en cuenta la jerarquía de intercambio, se define un mecanismo de “refinamiento” de soluciones. El AGII realiza la búsqueda en múltiples codificaciones, concentradas en diferentes áreas del espacio de soluciones. Los nodos de baja resolución trabajan en un espacio proporcionalmente menor, hallando soluciones “adecuadas” rápidamente, que se inyectan a los nodos de alta resolución, para orientarlos en su búsqueda.

El enfoque permite aplicar una técnica Divide & Conquer, no garantizada en los AGP homogéneos que en general producen soluciones pertenecientes a varios subespacios de búsqueda. El AGII procesa los diferentes niveles de resolución en paralelo y reduce el riesgo de búsqueda en intervalos incorrectos evitando la convergencia prematura.

En su reseña de 1996, Schwehm [66] discute los tipos de AGP independizándose de implementaciones para hardware especial. El resultado es una categorización precisa de los AGP, basada en la organización estructural de la población, que explica la correspondencia con los modelos de otras taxonomías estudiadas [40] [48] [36] [28]:

- El *modelo global* implementa el AG serial sobre una plataforma paralela. La selección proporcional es un cuello de botella de la implementación, identificándose variantes de acuerdo al mecanismo para lidiar con dicho problema. Las versiones con selección secuencial omiten paralelizar la selección y son implementables como un modelo maestro-esclavo con la población almacenada en memoria compartida o en el proceso maestro, siendo útiles cuando la evaluación de la función de fitness es costosa. Las versiones con selección paralela distribuyen el mecanismo de selección por torneo, el cálculo del fitness absoluto para la selección proporcional o la ordenación para la selección por rangos. Una vez seleccionados, subsiste el problema de obtener los individuos para el cruzamiento sin conflictos de memoria o cuellos de botella en las comunicaciones.
- El *modelo regional* es una extensión al modelo clásico de AG. La población se divide en *demes* que ejecutan un AG independientemente. Los mecanismos globales de selección, cruzamiento y reemplazo se reemplazan por mecanismos locales y se agrega el operador de migración para intercambio de material genético entre las poblaciones. Se establecen variantes de acuerdo a la estructura y número de población de las regiones, así como las topologías de migración y los parámetros que la controlan –estrategia, tasa y frecuencia de migración. Estos parámetros determinan cuánto se acerca una instancia del modelo regional al modelo global –topología de migración completa, alta tasa y frecuencia de migración– o al modelo local –topología de migración rígida.
- El *modelo local* proporciona una estructura espacial a la población del AG. Los mecanismos de selección, reproducción y reemplazo se llevan a cabo localmente, dentro de una vecindad definida por proximidad espacial entre los individuos. La topología de la vecindad y su tamaño (radio) son los principales parámetros espaciales del modelo. Las estrategias de selección y reemplazo local determinan la presión de selección, usualmente menor en este modelo dadas las limitaciones impuestas por la estructura espacial. Las estrategias usuales de cruzamiento local utilizan el valor de fitness proporcional local o un rango proporcional, o selección por torneo local, seleccionando dos individuos de la vecindad para reproducción o un único individuo que forzosamente se apareará con el individuo central. La estrategia de reemplazo estándar para este modelo es el reemplazo generacional del individuo que ocupa el lugar central de la vecindad, ya que el reemplazo elitista no es utilizable como técnica local, al necesitar conocimiento de información 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.

#### 4. 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. Esta etapa se caracteriza por trabajos de investigación que tratan de independizarse del hardware subyacente a los AGP y ofrecer taxonomías genéricas, que contemplen la mayoría de los modelos existentes.

Dada la difusión de las arquitecturas distribuidas de bajo costo (clusters de PCs o workstations), crece el estudio de los modelos distribuidos, aunque continúan desarrollándose implementaciones para equipos de memoria compartida.

Tomassini, en sus reseñas sobre AG [74] y sobre AGP [75], hace referencia a varios modelos de paralelismo, basados en la idea de la existencia de jerarquías de aplicación de las técnicas de distribución.

El modelo de *paralelismo global* surge de la aplicación del paralelismo a nivel de la función de fitness.

La aplicación del paralelismo a nivel de la población permite implementar el *modelo de islas*, poblaciones semi-independientes que ejecutan el AG serial, con el agregado de migraciones de individuos.

Aplicando el paralelismo a nivel de individuos, se obtienen los AG *celulares*. La terminología *celular* es adoptada por la similitud con los autómatas celulares con reglas de transición estocásticas [82] [72] [73]. Se menciona la adecuación de este modelo a las arquitecturas SIMD, que implementan comunicaciones eficientes por hardware.

Tomassini presenta una clasificación que considera el comportamiento espacial y temporal de AGP, reconociendo versiones sincrónicas y asincrónicas, de acuerdo a la coordinación entre poblaciones mediante las comunicaciones. Los AGP híbridos –que combinan modelos de paralelismo– y los no uniformes –que usan poblaciones con diferentes codificaciones, mecanismos y/o parámetros–, se engloban en una categoría de modelos *No estándar*.

Venkateswaran, Obradovic y Raghavendra proponen un modelo de AG *cooperativos* [79], un enfoque intermedio entre el modelo global y el modelo de islas. Múltiples procesos ejecutan un AG global con la totalidad de la población en paralelo, trabajando simultáneamente sobre diferentes áreas del espacio de soluciones, mejorando las probabilidades de obtener buenas soluciones. El modelo es heterogéneo, ya 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 diseño maestro-esclavo sobre una plataforma distribuida y utilizado para resolver problemas de asignación, reportándose resultados óptimos para pequeños problemas y mejoras de eficiencia respecto al AG serial para problemas con relación tiempo de comunicación/tiempo de ejecución pequeña.

En sus trabajos [83] [84] [85] [86] Whitley et al. analizan exhaustivamente el modelo de islas con subpoblaciones y migración. En sus trabajos estudian detalladamente el intervalo de migración y el tamaño de la población migrada, parámetros que controlan el mecanismo de intercambio. Conjuntamente, se analizan diferentes representaciones para problemas de optimización linealmente separables y las mejoras obtenidas con respecto a la calidad de la solución al utilizar el modelo de islas. 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 de Cantú-Paz [19] [20] [21] [22] [23], culminaron con la publicación de un completo texto que reúne análisis teóricos, aspectos de implementación y estudios de eficiencia para AGP [24]. Su clasificación de AGP extiende los modelos clásicos de Grefenstette [40], de modo similar a propuestas ya comentadas [4] [48] [7] [74]. Tomando en cuenta la distribución de la población y de los operadores evolutivos, reconoce cuatro categorías:

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

En [20] Cantú-Paz estudia los AGP maestro-esclavo, destacando las ventajas de este enfoque con respecto a otros modelos: utilizan el mismo mecanismo de exploración del espacio de soluciones que los AG seriales, siendo aplicables las consideraciones de diseño deducidas teórica y experimentalmente para éstos; son sencillos de implementar y proporcionan mejoras de performance para problemas con funciones de fitness complejas de evaluar. Analizando los tiempos de cálculo y de comunicación de información entre procesos para el modelo sincrónico, obtiene cotas inferiores para la eficiencia del modelo, número óptimo de procesadores y speedup. El tiempo de cálculo del proceso evolutivo depende del tamaño de la población, cuya reducción implica una pérdida potencial de la calidad de la solución. El tiempo de evaluación de la función de fitness disminuye al aumentar el número de procesos esclavos, pero el overhead de comunicación crece al aumentar la cantidad de procesos.

Cantú-Paz argumenta que 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. La paralelización de estas últimas no es usual ni útil, ya que al ser simples y necesitar información global, la mejora de performance obtenida se pierde en tareas de comunicación y sincronización.

En la versión asincrónica, cada esclavo envía los resultados tan pronto como finaliza y el maestro genera nuevos individuos y los envía a los esclavos, sin demoras. Esta versión tiene el potencial de ser más eficiente que la sincrónica, en especial si los tiempos de evaluación de la función de fitness no son constantes para todos los individuos. Como contrapartida, los AGP maestro-esclavo asincrónicos son algo más complejos de implementar, y el no-determinismo de los procesos asincrónicos los hacen más difíciles de analizar que su modelo sincrónico.

En su línea de trabajo de aplicación del paralelismo a los AG *desordenados*, Lamont, Gates y Merkle proponen una implementación utilizando la biblioteca MPI en su trabajo de 1997 [44]. El modelo, basado en el algoritmo *fmGA* (*fast messy genetic algorithm*) de Goldberg et al. [32], explota 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 [45].

Alba y Troya [7] analizan la importancia del sincronismo en las migraciones para AGP con poblaciones distribuidas. En su trabajo, identifican a los AGP de grano fino y de grano grueso como subclases de un modelo único compuesto por algoritmos comunicados y proponen denominarlos AG *distribuidos* y *celulares* respectivamente, argumentando que la granularidad de un problema paralelo refiere a la relación entre trabajo computacional y comunicación, mientras que los modelos de AGP se diferencian en el modo que estructuran la población y no en su granularidad. 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 [6] [8] [9], Alba y Troya confirman que es usual que los segundos superen en eficiencia a los primeros.

Baraglia y Perego [13] estudiaron implementaciones de AGP de grano fino sobre multiprocesadores con conexión de hipercubo. En su breve taxonomía discriminan entre el modelo *centralizado* con población global y los modelos *distribuidos* de *grano fino* y de *grano grueso*. Ofrecen un análisis comparativo entre AGP de grano fino y de grano grueso implementados sobre un multiprocesador nCUBE 2 de 128 nodos, 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 [41]. 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 1999, Sprave [68] 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. Sprave propone modelar las estructuras de población mediante hipergrafos, estudiando los modelos de migración y de grano fino. El entorno definido se presenta como una poderosa herramienta para el cálculo de medidas de la presión de selección, como el 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 sature la totalidad de la población, asumiendo que el único operador aplicado es el de selección.

El paralelismo también se ha aplicado a los AG de estado estacionario. Berger, et al. [14] presentan el modelo *PHGA* con poblaciones solapadas y lo aplican a problemas de Ruteo, combinando la potencialidad de la evolución paralela y la relajación parcial con constantes temporales para explorar exhaustivamente el espacio de búsqueda. Incluyen innovaciones a los operadores de evolución del AG estándar, incorporando conceptos de técnicas de optimización emergentes como las heurísticas de inserción o las colonias de hormigas. Los resultados presentados indican que el modelo *PHGA* logra mejorar la intensidad y la diversificación de la búsqueda, reportándose como un algoritmo competitivo y de mejor performance que las metaheurísticas conocidas para problemas de Ruteo.

Davidson y Rasheed [26] estudian los efectos del paralelismo global aplicado a los AG de estado estacionario. La investigación empírica con su algoritmo GADO (*Genetic Algorithm for Design Optimization* [64]) prueba que el modelo maestro-esclavo permite obtener mejoras de significativas de eficiencia sin afectar la convergencia del AG.

En 1999, Nowostawski y Poli [61] presentan una taxonomía unificada que reúne un amplio número de modelos propuestos en la literatura. Su trabajo sigue un enfoque globalizador, en un intento de que su propuesta comprenda a la totalidad de las clasificaciones elaboradas hasta ese momento.

Ocho son las categorías identificadas por Nowostawski y Poli, quienes distinguen entre los siguientes modelos :

- *modelo maestro-esclavo*, con sus versiones sincrónicas y asincrónicas, de acuerdo al criterio adoptado para coordinar las poblaciones antes de iniciar una nueva generación.
- *modelo distribuido de subpoblaciones con migración*. El operador de migración permite discriminar entre variantes de acuerdo a sus parámetros: topología, tasa y frecuencia de migración y esquema de selección y reemplazo de individuos migrados. El extinto modelo *stepping stone* es citado como un caso particular de esta categoría. En los extremos de este enfoque están los AGP de *grano fino* (adecuados para arquitecturas SIMD de memoria compartida) y de *grano grueso* (usualmente implementados en modelos de máquinas paralelas MIMD distribuidas). El principal inconveniente de este modelo consiste en la dificultad de implementar soluciones escalables para grandes problemas, lo cual generalmente conduce al diseño de modelos híbridos que permitan un uso más inteligente del hardware paralelo disponible, introduciendo jerarquías de procesamiento.
- *modelo de poblaciones estáticas solapadas, sin migración*. Al no contar con un operador de migración, el intercambio de información entre poblaciones se realiza a través de las áreas solapadas, formadas por individuos que pertenecen a más de un deme. La ausencia de comunicaciones explícitas hace a este modelo adecuado para implementar en arquitecturas SIMD, realizando los intercambios a través de memoria compartida.
- *AG masivamente paralelos*. Constituyen 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.
- *modelo de poblaciones dinámicas, con solapamiento*, originalmente presentado por Nowostawski [43] [59] [61] y definido en detalle en [60]. Es una propuesta híbrida que combina la evaluación distribuida de fitness del modelo maestro-esclavo y el modelo de poblaciones estáticas con solapamiento. En cada generación, los demes se crean dinámicamente tan pronto como existan suficientes individuos. Este mecanismo de reorganización es el encargado del intercambio de información entre poblaciones, eliminando el efecto de los individuos más lentos en el modelo maestro-esclavo. El modelo corresponde a un algoritmo MIMD maestro-esclavo asincrónico totalmente escalable, adecuado para la ejecución en máquinas distribuidas o masivamente paralelas.

- *AGP de estado estacionario*, cuyo mecanismo de actualización continua de la población permite la aplicación usual de las técnicas de paralelismo. Los mecanismos de selección y reemplazo deben ser secuenciales, ya que los individuos están en la sección crítica del algoritmo, pero los restantes operadores pueden paralelizarse.
- *AG desordenados paralelos*, cuyas características permiten aplicar técnicas específicas de paralelismo de datos, que fueron comentadas al analizar los trabajos de Merkle, Lamont y Gates [51] [44] [45].
- *modelos híbridos*, que combinan diferentes modelos para paralelizar AG. Usualmente definen jerarquías entre poblaciones, donde se aplican enfoques diferentes, combinando las ventajas de varios modelos de paralelismo. Los modelos híbridos son analizados en detalle por Cantú-Paz [24], bajo el nombre de *modelos jerárquicos*.

## 5. 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 es manifiesto.

La referencia inmediata al estado actual de investigación en el área de los AGP es el texto de Cantú-Paz [24]. Sin embargo, la clasificación en él propuesta no contempla algunos modelos de paralelismo con características propias, y la categoría de *modelos híbridos* reúne una amplia gama de enfoques, y es susceptible de un refinamiento. Por los motivos expuestos, la clasificación de Nowostawski y Poli emerge como una alternativa más comprehensiva y específica. A su vez, se presenta como una clasificación más flexible y útil, incluyendo categorías para modelos no tradicionales, como los AGP *desordenados* o los AGP *de estado estacionario*, ausentes en la taxonomía clásica

Es posible vincular la taxonomía de Nowostawski y Poli con los modelos de propuestos por otros investigadores.

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

- *modelos de paralelismo global* de Gordon & Whitley [34], Schwehm [66], y Tomassini [74], [75]
- *modelo de paralelismo estándar* Adamidis [3]
- *modelo micrograno* de Lin et al. [48].
- *modelo de implementación centralizada* por Bianchini & Brown [16] y Baraglia & Perego [13]

El *modelo distribuido de subpoblaciones con migración* incluye a los siguientes enfoques:

- el *modelo PGA* de Petty & Leuze [62]
- el *modelo de islas* de Gorges-Schleuter [35], Gordon & Whitley [34], Whitley [82] y Tomassini [74] [75]
- el *modelo forkingGA* de Tsutsui & Fujimoto [76]
- el *modelo sin migración* de Shonkwiler [67]
- el *modelo heterogéneo* de Tanese [71]
- el *modelo distribuido de grano fino* de Maruyama, Hirose & Konagaya [50]
- el *modelo de implementación distribuida* de Bianchini & Brown [16]
- el *modelo de descomposición* de Adamidis [3].
- el *modelo de AGP de grano grueso* de Lin et al. [48] y de Baraglia & Perego [13]
- el *modelo regional* de Schwehm [66].
- el *modelo de AG distribuidos* de Alba & Troya [8]

El *modelo de poblaciones estáticas solapadas sin migración* corresponde al modelo de grano fino original de Robertson [64] e incluye las siguientes propuestas:

- el *modelo de AGP de grano fino* de Mühlenbein [53], Manderick & Spiessen [49] y Baraglia & Perego [13]
- el *modelo de vecindades* de Gorges-Schleuter [35].
- el *modelo de polinización* de Goldberg [30].
- el *modelo de AGP de grano fino* de Lin et al. [48].
- el *modelo local* de Schwehm [66].
- el *modelo estocástico* de Tomassini [74] [75].

La categoría de AG masivamente paralelos incluye a los modelos :

- *modelo mpdGA* de Baluja [11] [12].
- *AGP celulares* de Gordon & Whitley [34], Tomassini [74] [75] y Alba & Troya [8].

El modelo de poblaciones dinámicas, con solapamiento es reciente y original de Nowostawski & Poli [43] [58] [59] [60]. 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 [46] [47].
- el *modelo estacionario maestro-esclavo* de Davidson & Rasheed [26].
- el *modelo híbrido multipoblación (PHGA)* de Berger, Barkaoui & Bräysy [14].

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

- el modelo de distribución de la evaluación de la función de fitness propuesto por Goldberg [33].
- el modelo de distribución de datos en la fase primordial, idea de Merke & Lamont [51].
- el modelo fmGA (*fast messy genetic Algorithm*) propuesto por Lamont, Gates & Merkle [45].

Los modelos híbridos engloban a aquellos que combinan enfoques y estrategias, y a las propuestas originales 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 [30].
- los *modelos de AG cooperativos* de Adamidis [4], [5] y Venkateswaran, Obradovic & Raghavendra [79].
- el modelo distribuido jerárquico con meta-optimización de parámetros de Munetomo et al. [56].
- el *modelo de implementación semidistribuido* con organización en clusters de Bianchini & Brown [16].
- el modelo distribuido con secciones compartidas de Hiroyasu, Miki & Watanabe [41].
- los denominados *modelos jerárquicos* por Cantú-Paz [24].

De acuerdo a la lista precedente, la propuesta de unificación de terminologías y taxonomías de modelos de AGP constituye una necesidad importante para los investigadores en esta área. La alternativa de modelos es variada, y determinar el adecuado para aplicar a un determinado problema de optimización no es una tarea sencilla sin tener un panorama de las características de los modelos, sus implementaciones y las ventajas y desventajas respecto al resto.

La intención de este trabajo ha sido presentar las principales propuestas, resumiendo algunos detalles relevantes de los artículos, textos y reseñas disponibles y enmarcándolos dentro de la categorización de Nowostawski & Poli.

## Referencias bibliográficas

- [1] D. Abramson, J. Abela, A Parallel Genetic Algorithm for Solving the School Timetabling Problem. Proceedings of the 15th Australian Computer Science Conference, 1992, Vol. 14 pp. 1-11
- [2] D. Abramson, G. Mills, S. Perkins, Parallelisation of a Genetic Algorithm for the Computation of Efficient Train Schedules. Proceedings of 1993 Parallel Computing and Transputers Conference, pp 139 – 149, 1993.
- [3] P. Adamidis, V. Petridis, Co-operating Populations with Different Evolution Behavior. Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), p. 188-191, Nagoya, Japan, 1996.
- [4] P. Adamidis, Parallel Evolutionary Algorithms: A Review. HERCMA'98, Atenas, Grecia, 1998.
- [5] P. Adamidis, Improving the performance of Parallel Genetic Algorithms with Co-operating Populations with Different Evolution behavior. Ph.D. Thesis, Aristotle Univ. of Thessaloniki, Grecia, 1998.
- [6] E. Alba, C. Cotta, J. Troya, Entropic and Real-Time Analysis of the Search with Panmictic, Structured, and Parallel Distributed Genetic Algorithms. Proceedings of the 1999 GECCO, Morgan Kaufmann, pp. 773, 1999.
- [7] E. Alba, J. Troya, A Survey of Distributed Genetic Algorithms. Complexity 4(4), pp.31-52, 1999.
- [8] E. Alba, J. Troya, An Analysis of Synchronous and Asynchronous Parallel Distributed Genetic Algorithms with Structured and Panmictic Islands. Proceedings of the 11th IPSP/SPDP99, 1999.
- [9] E. Alba, J. Troya, Analyzing Synchronous and Asynchronous Parallel Distributed Genetic Algorithms. Future Generation Computer Systems, 17(4) pp. 451-465, enero 2001.
- [10] A. Balakrishnan, Parallel Genetic Algorithms, Premature Convergence and the nCUBE AComs Project, 1993.
- [11] S. Baluja, A Massively Distributed Parallel Genetic Algorithm. CMU-CS-92-196R, Carnegie Mellon, 1992.
- [12] S. Baluja, Structure and Performance of Fine Grain Parallelism in Genetic Search. Proceedings of the Fifth International Conference (ICGA93), Morgan Kaufmann Publishers, San Mateo, CA., pp. 155-162, 1993.
- [13] R. Baraglia, R. Perego, Parallel Genetic Algorithms for Hypercube Machine. Proceedings of Third International Conference for Vector and Parallel Processing - VECPAR 98, Portugal, 1998, pp. 749-761, 1998.
- [14] J. Berger, M. Barkaoui, O. Bräysy, A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows. Defense Research Establishment Valcartier, Canada, 2001.
- [15] A. Bethke, Comparison of genetic algorithms and gradient based optimizers on parallel processors : Efficiency of use of precessing capacity. Technical Report No 197, Ann Arbor, University of Michigan. Citado en [24]
- [16] R. Bianchini, C. M. Brown, Parallel Genetic Algorithms on Distributed-Memory Architectures. Technical Report 436, University of Rochester. Computer Science Department, 1993.
- [17] R. Bianchini, C. Brown, M. Cierniak, W. Meira, Combining Distributed Populations and Periodic Centralized Selections in Coarse-Grain Parallel Genetic Algorithms. Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms 95, Abril 1995.
- [18] W. Bossert, Mathematical optimization: are there abstract limits on natural selection? Mathematical Challenges to the Neo-Darwinian intepretation of evolution, Wistar Inst. Press, 5, pp. 35-46, 1967. Citado en [24]
- [19] E. Cantú-Paz, A Summary of Research in Parallel Genetic Algorithms. IlliGAL, 1998.
- [20] E. Cantú-Paz Designing efficient master-slave parallel genetic algorithms. Genetic Programming: Proceedings of the Third Annual Conference. pp. 455, San Francisco, 1998.
- [21] E. Cantú-Paz, On the scalability of parallel genetic algorithms. Evolutionary Computation. 7(4), 429-449, 1999.
- [22] E. Cantú-Paz Markov chain models of parallel genetic algorithms. IEEE Transactions on Evolutionary Computation.4(3), 216-226, 2000.

- [23] E. Cantú-Paz, Selection intensity in genetic algorithms with generation gaps. GECCO-2000: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 911-918, San Francisco, 2000
- [24] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Kluwer Academic Publishers, 2001
- [25] L. Davis. Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.
- [26] B. Davison, K. Rasheed, Effect of Global Parallelism on the Behavior of a Steady State Genetic Algorithm for Design Optimization. Proceedings of the Congress of Evolutionary Computation CEC99 p. 534-541, 1999.
- [27] K. DeJong. Analysis of the behaviour of a class of genetic adaptive systems. PhD tesis Univ of Michigan, 1975.
- [28], M. Dorigo, A. Bertoni, Implicit Parallelism in Genetic Algorithms, Artificial Intelligence (61), 307-314, 1993.
- [29] T. Fogarty, R. Huang, Implementing the Genetic Algorithm on Transputer Based Parallel Processing Systems. Parallel Problem Solving from Nature 1990, pp. 145-149.
- [30] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, NY, 1989.
- [31] D. Goldberg Sizing populations for serial and parallel genetic algorithms. Proceedings of the Third International Conference on Genetic Algorithms, pp. 70-79, 1989.
- [32] D. Goldberg, B. Korb, K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. Complex Systems, 3:493--530, 1990.
- [33] D. Goldberg, K. Deb, H. Kargupta, G. Harik. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. Proceedings of the Fifth International Conference on Genetic Algorithms, pp 56-64., 1993
- [34] V. Gordon, D. Whitley, Serial and Parallel Genetic Algorithms as Function Optimizers. Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 177-183, 1993.
- [35] M. Gorges-Schleuter, ASPARAGOS: An asynchronous parallel genetic optimization strategy. Proceedings of the Third International Conference on Genetic Algorithms, pp.422-427, 1989.
- [36] M. Gorges-Schleuter. Explicit parallelism of genetic algorithms through population structures. Proceedings of 1st PPSN'90, pp. 150-159, 1990.
- [37] M. Gorges-Schleuter, Comparison of Local Mating Strategies in Massively Parallel Genetic Algorithms. Proceedings of 2nd PPSN, pp. 559-568, 1992.
- [38] M. Gorges-Schleuter, Asparagos96 and the traveling salesman problem. Proceedings of 1997 IEEE International Conference on Evolutionary Computation, pp. 171-174. IEEE, Abril 1997.
- [39] M. Gorges-Schleuter. A Comparative Study of Global and Local Selection in Evolution Strategies. PPSN V Parallel Problem Solving from Nature, pp. 367-377, 1998.
- [40] J. Grefenstette. Parallel adaptive algorithms for function optimization. Technical Report CS-81-19, Computer Science Department, Vanderbilt University, Nashville, TN, 1981. Citado en [30].
- [41] T. Hiroyasu, M. Miki, S. Watanabe. Distributed Genetic Algorithms with a New Sharing Approach in Multiobjective Optimization Problems. IEEE Proceedings Congress on Evolutionary Computation, pp.69-76, 1999.
- [42] J. Holland, Outline for a logical theory of adaptive systems. Journal of the Association for Computer Machinery, 1963, 3, pp.297-314. Citado en [30]
- [43] H. Kwasnicka, M. Nowostawski. The Search Engine for Information Systems Based on Parallel Genetic Algorithm. Proceedings 2nd Int. Conference on Parallel Processing & Applied Mathematics, pp. 442-451, 1997.
- [44] G. Lamont, G. Gates & L. Merkle. A MPI Implementation of the Fast Messy Genetic Algorithm. Proceedings of the Intel Supercomputer Users Group Conference (ISUG'97), 1997.
- [45] G. Lamont, G. Gates & L. Merkle. Scalability of a MPI-Based Fast Messy Genetic Algorithm, Proceedings of ACM Symposium on Applied Computing (SAC'98), 1998.
- [46] D. Levine. A Genetic Algorithm For The Set Partitioning Problem. Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, Morgan Kaufmann, 1993, pp. 81--487
- [47] D. Levine. A Parallel Genetic Algorithm for the Set Partitioning Problem, PhD Tesis, Illinois IT, 1994.
- [48] S.C. Lin, W.F. Punch, E.D. Goodman Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach. Proceedings of the Sixth IEEE Parallel & Distributed Processing, IEEE Press, pp. 28-37, 1994
- [49] B. Manderick, P. Spiessens. Fine-Grained Parallel Genetic Algorithms. Proceedings of the Third International Conference on Genetic Algorithms, pp. 428-433, 1989.
- [50] T. Maruyama, T. Hirose, A. Konagaya. A Fine-Grained Parallel Genetic Algorithm for Distributed Parallel Systems. Proceedings of the Third International Conference on Genetic Algorithms, pp 184-190, 1993.
- [51] L. Merkle, G. Lamont. Comparison of Parallel Messy Genetic Algorithms Data Distribution Strategies. Proceedings of Fifth International Conference on Genetic Algorithms, pp. 191-198, 1993.
- [52] H. Mühlenbein, M. Gorges-Schleuter, O. Kramer. Evolution algorithms in combinatorial optimization. Parallel Computations, 7, pp. 65-85, 1988.
- [53] H. Mühlenbein. Parallel genetic algorithms, population genetic and combinatorial optimization. Proceedings of the Third International Conference on Genetic Algorithms, pp 416-421, Arlington, 1989.
- [54] H. Mühlenbein, Evolution in Time and Space - The Parallel Genetic Algorithm. Foundations of Genetic Algorithms, Morgan-Kaufman, pp. 316-337, 1991.
- [55] M. Munetomo, Y. Takai, Y. Sato. An Efficient Migration Scheme for Subpopulation-Based Asynchronously Parallel Genetic Algorithms. Proceedings of the 5th International Conference on Genetic Algorithms, pp. 649, 1993.
- [56] M. Munetomo, M. Takahashi, Y. Takai, Y. Sato. A Cooperative Search Strategy Using Hierarchical Genetic

- Algorithms and Its Implementation on the UNIX-Network. IPSJ SIGNotes Contents Artificial Intelligence No.091
- [57] S. Nasmachnow, Aplicación de las técnicas de procesamiento paralelo a la implementación de Algoritmos Genéticos. Technical Report INCO 06-02, Universidad de la República, Uruguay, 2002.
- [58] M. Nowostawski. Parallel Genetic Algorithms in Geometry Atomic Cluster Optimization and Others Applications. M.S. thesis, School of Computer Science, The university of Birmingham, UK, 1998.
- [59] M. Nowostawski. Parallel Genetic Algorithms and Sequencing Optimisation, Mini-project report, 1998. Online (<http://marni.otago.ac.nz/~mariusz/gzipped/rep2r.ps.gz>), consultado junio 2002.
- [60] M. Nowostawski, R. Poli. Dynamic Demes Parallel Genetic Algorithm. Proceedings of Third International Conference on Knowledge-based Intelligent Information Engineering Systems KES'99.
- [61] M. Nowostawski, R. Poli. Parallel Genetic Algorithms Taxonomy. Proceedings of Third International Conference on Knowledge-based Intelligent Information Engineering Systems KES'99
- [62] C. Pettey, M. Leuze, J. Grefenstette. Genetic Algorithms and their implementation. Proceedings of the Second International Conference on Genetic Algorithms, Morgan Kauffmann Publishers, pp. 155-161, 1987.
- [63] C. Pettey, M. Leuze, J. Grefenstette. A Theoretical Investigation of a Parallel Genetic Algorithm. Proceedings of the Third International Conference on Genetic Algorithms, 1989, pp. 398-405, 1989.
- [64] K. Rasheed. GADO: A genetic algorithm for continuous design optimization. Technical Report DCS-TR-352. Department of Computer Science, Rutgers University. New Brunswick, NJ. Ph.D., 1998.
- [65] G. Robertson, Parallel Implementation of Genetic Algorithms in a Classifier System. Proceedings of the Second International Conference on Genetic Algorithms, pp. 140-147, 1987.
- [66] M. Schwehm Parallel Population Models for Genetic Algorithms. Foundations of Genetic Algorithms Workshop (FOGA4), San Diego, California, 1996.
- [67] R. Shonkwiler. Parallel Genetic Algorithms. Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kauffman, 1993, pp. 658.
- [68] J. Sprave. A Unified Model of Non-Panmictic Population Structures in Evolutionary Algorithms. Technical Report ISSN 1433-3325 Universität Dortmund, 1999.
- [69] T. Starkweather, D. Whitley, K. Mathias, Optimization Using Distributed Genetic Algorithms. Parallel Problem Solving from Nature, p. 176-185, Springer-Verlag (Berlin), 1991.
- [70] R. Tanese. Distributed Genetic Algorithms for Function Optimization. Ph. D. Thesis, Department of Electrical Engineering and Computer Science, University of Michigan, 1989.
- [71] R. Tanese. Distributed Genetic Algorithms. Proceedings of the Third International Conference on Genetic Algorithms, Kauffman, San Mateo, 1989, pp 434-439.
- [72] M. Tomassini, Massively parallel evolutionary algorithms. Proc. 2nd Connection Machine User Meeting, 1993.
- [73] M. Tomassini. The Parallel Genetic Cellular Automata: Application to Global Function Optimization. Artificial Neural Nets and Genetic Algorithms pp. 385-391, 1993.
- [74] M Tomassini. A Survey of Genetic Algorithms. Annual Reviews of Computational Physics, pp. 87-118, 1995.
- [75] M. Tomassini. Parallel and Distributed Evolutionary Algorithms: A Review. Evolutionary Algorithms in Engineering and Computer Science, pp. 113-133, 1999.
- [76] S. Tsutsui, Y. Fujimoto. Forking Genetic Algorithm with Blocking and Shrinking Modes (fGA). Proceedings of the 5th International Conference on Genetic Algorithm (ICGA), pp. 206-213, 1993.
- [77] S. Tsutsui, Y. Fujimoto. Extended Forking Genetic Algorithm for Order Representation (o-fGA). Proceedings of the First IEEE International Conference on Evolutionary Computation (ICEC'94), Vol. 1, pp. 170-175, 1994.
- [78] S. Tsutsui, Y. Fujimoto, A. Ghosh. Forking GAs: GAs with Search Space Division Schemes. Evolutionary Computation, Vol. 5, No. 1, pp. 61-80, 1997.
- [79] R. Venkateswaran, Z. Obradovic, C.S.Raghavendra. Cooperative Genetic Algorithm for Optimization Problems. Proceedings of the 2nd Online Workshop on Evolutionary Computation, pp. 49-52, 1996
- [80] D. Whitley. The GENITOR Algorithm and Selection Pressure : Why Rank-Based Allocation of Reproductive Trials is Best. Proceedings of the Third International Conference on Genetic Algorithms, pp 116-121, 1989.
- [81] D. Whitley, T. Starkweather. GENITOR II: a Distributed Genetic Algorithm. Journal of Experimental and Theoretical Artificial Intelligence 2, pp 189-214, 1990.
- [82] D. Whitley. Cellular Genetic Algorithms. Proceedings of the Fifth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, 1993, pp. 658.
- [83] D. Whitley, R. Beveridge, C. Guerra, C. Graves. Messy Genetic Algorithms for Subset Feature Selection. International Conference on Genetic Algorithms, 1997.
- [84] D. Whitley, S. Rana, R. Heckendorn. Island Model genetic Algorithms and Linearly Separable Problems. Evolutionary Computing, AISB Workshop 1997, pp. 109-125
- [85] D. Whitley, S. Rana, R. Heckendorn. The Island Model Genetic Algorithm: On Separability, Population Size and Convergence. Special Issue on Evolutionary Computing Vol. 7, N° 1, 1999, pp. 33-47
- [86] D. Whitley, S. Rana, R. Heckendorn Representation Issues in Neighborhood Search and Evolutionary Algorithms. Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Pp. 39-57, 1998.
- [87] D. Whitley, J. Kauth. GENITOR: A Different Genetic Algorithm. Technical Report CS-88-101, Colorado State University, Department of Computer Science, Fort Collins, 1998.