

Capítulo 1

Introducción

Los algoritmos con inspiración biológica, también conocidos como algoritmos bioinspirados, han adquirido gran importancia dentro del área de la inteligencia artificial debido a que han demostrado ser exitosos en la solución de ciertos problemas complejos de aprendizaje de máquina, reconocimiento de patrones, detección de fallas y optimización. El sistema inmune artificial (SIA), inspirado por algunos procesos observados principalmente en el sistema inmune biológico de los mamíferos, es un área emergente en los sistemas computacionales adaptativos que incluye algoritmos que han sido empleados satisfactoriamente, entre otros, a problemas de optimización global como combinatoria, tratándose de una técnica heurística relativamente nueva y que ha resultado altamente competitiva. En el sistema inmune biológico, el principio de selección clonal es uno de los procesos nativos más estudiados y que es abstraído directamente como un algoritmo particular del SIA, utilizándose principalmente para resolver problemas de optimización numérica y reconocimiento de patrones.

Al igual que sucede con los algoritmos evolutivos, el principio de selección clonal del SIA es *poblacional*, lo que significa que manipula simultáneamente un conjunto de soluciones potenciales del problema, implicando en la mayoría de los casos un tiempo de ejecución elevado y un amplio uso de la memoria de datos que se traduce en un alto costo computacional. Una alternativa para reducir tal costo ha sido el diseño de algoritmos con poblaciones extremadamente pequeñas (micro-algoritmos) que al trabajar con pocos individuos realizan menos evaluaciones de la función objetivo utilizando por ende un menor espacio en la memoria de datos del sistema de cómputo. Cumpliendo con una elección correcta de los parámetros sensibles en este tipo algoritmos, por ejemplo el tamaño de la población o la probabilidad de

mutación, entre otras variables, es posible acelerar la convergencia en comparación a las versiones extendidas de los mismos.

El principio de selección clonal se caracteriza por aumentar el tamaño de la población durante la etapa de clonación, lo que exige la manipulación de más individuos que los existentes al inicio del algoritmo, y sólo permite aplicar operadores de mutación (no hay operador de cruza), por lo que mantener la diversidad de la población y asegurar la convergencia del algoritmo, son retos que hacen interesante el estudio de esta heurística bioinspirada.

En este trabajo de tesis se presenta un algoritmo basado en el principio de selección clonal del sistema inmune artificial, nombrado *micro-SIA* (micro-sistema inmune artificial), que utiliza una población reducida en su número de individuos (anticuerpos, en la terminología propia del SIA) y que fue diseñado primeramente en software para solucionar problemas de optimización numérica y que posteriormente se implementó intrínsecamente en hardware. Las versiones en software del micro-SIA se presentan en las dos vertientes abordadas en la literatura especializada para optimización de un solo objetivo: sin manejo de restricciones y con manejo de espacios restringidos.

Se realizaron dos arquitecturas del micro-SIA propuesto, la primera en un microcontrolador comercial y la segunda en un dispositivo de lógica reconfigurable (FPGA). El *hardware evolutivo*, llamado así por ser una migración de la versión en software del algoritmo bioinspirado a una versión en alguna plataforma en hardware, puede ayudar a solucionar los problemas de diseño de circuitos, de tolerancia a fallos y adaptación a entornos cambiantes. Lo anterior se debe a que hardware evolutivo no usa reglas de diseño rígidas, sólo se guía por el comportamiento del circuito y por lo tanto el espacio de soluciones sobre el que trabaja es mayor, además un diseño complejo permite que el sistema se rediseñe de manera autónoma, cuando se presente una falla o bien cuando las características del entorno varíen.

1.1 Algoritmos evolutivos y bioinspirados

Las técnicas heurísticas con inspiración biológica, también referidas en la literatura especializada como algoritmos evolutivos y bioinspirados [10, 11, 12] son procedimientos de búsqueda, optimización y aprendizaje, cuyo modelo se obtiene del mecanismo de la selección natural y las teorías de la evolución, basadas en el paradigma *Neo-Darwiniano* que aplica sobre esquemas poblacionales. Estas técnicas son muy populares, debido principalmente a su alto poder exploratorio y a su paralelismo implícito.

El Neo-Darwinismo es el conjunto de teorías que explican el origen de las especies en el planeta y cómo sus procesos de reproducción, mutación, competencia y selección permiten la adaptación y evolución de las especies; las teorías involucradas en este modelo son: el *seleccionismo de Weismann*, el *origen de las especies* formulado por Darwin y la *genética de Mendel* [10]. Todos estos algoritmos tienen en común el hecho de utilizar una población o conjunto de soluciones potenciales, generada comúnmente de manera aleatoria, y someterla a un proceso iterativo utilizando diferentes esquemas, operadores y estrategias en función del tipo de algoritmo.

En la particularidad de los algoritmos evolutivos, los operadores genéticos básicos son tres [11]: la *selección*, la *cruza* (reproducción o recombinación) y la *mutación*. La selección consiste de un mecanismo (puede ser probabilístico o determinista) que permite elegir a los individuos que fungirán como padres de la siguiente generación o iteración. La crusa se refiere al intercambio de información (material genético) entre dos padres que han sido seleccionados con base en su aptitud de acuerdo a una función objetivo. El último de los operadores listados, la mutación, se encarga de realizar pequeñas perturbaciones o cambios mínimos a los individuos recién creados para la nueva población con la finalidad de explorar zonas del espacio de búsqueda que la crusa pudiera no alcanzar, manteniendo la diversidad de los individuos.

La secuenciación de los pasos implicados en un algoritmo evolutivo estándar se lista en el algoritmo mostrado en la figura 1.1.

1. *Inicialización*: Población inicial generada aleatoriamente.
2. *Generación*: Aplicar los siguientes procedimientos evolutivos de adaptación,
 - 2.1. *Selección*: Se seleccionan los individuos que sobrevivirán y se reproducirán, de acuerdo a su aptitud con base en la función objetivo.
 - 2.2. *Reproducción y Variación Genética*: Se crean nuevos individuos por medio de la recombinación y/o introduciendo variaciones genéticas (mutación) en individuos seleccionados.
 - 2.3. *Evaluación de Aptitud*: evaluar nuevamente la aptitud de cada individuo modificado.
3. *Ciclo*: Repetir el paso 2 hasta que se alcance el criterio de convergencia.

Figura 1.1. Secuenciación de un algoritmo evolutivo estándar.

En la figura 1.2 se describe gráficamente el comportamiento y la ejecución del algoritmo evolutivo estándar, obsérvese que los operadores genéticos modifican las características de los individuos en el modelo poblacional.

Existen básicamente dos etapas de evaluación de la función objetivo, la primera en la *Selección* y posteriormente en la *Reselección* después de haber aplicado los operadores genéticos.

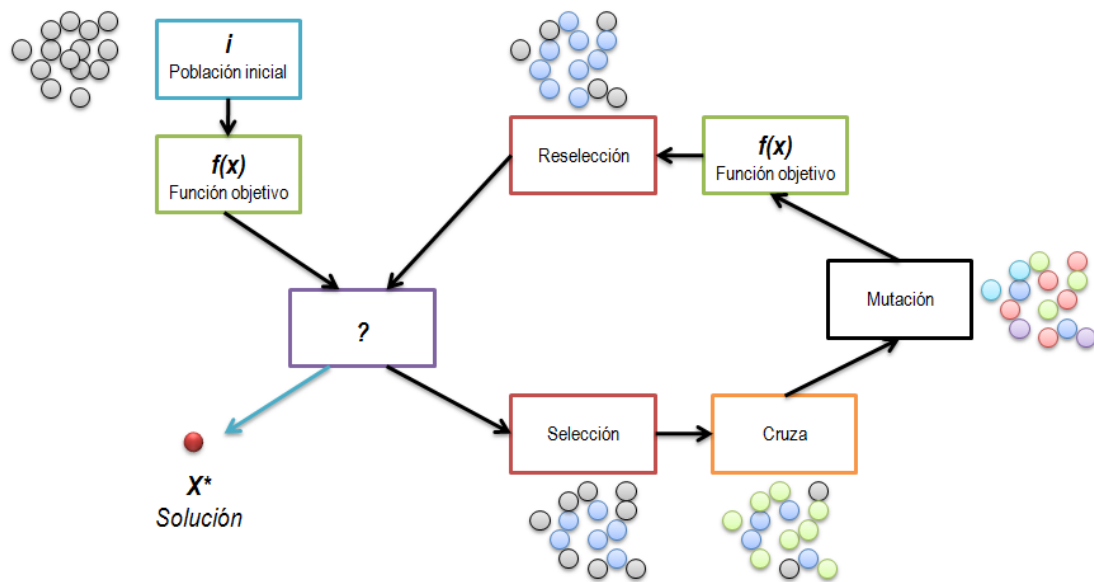


Figura 1.2. Comportamiento de un algoritmo evolutivo estándar.

Han surgido diferentes disciplinas que intentan imitar a través de medios computacionales, los procesos antes indicados, con la iniciativa de resolver problemas complejos *NP*, lineales y no lineales. Una de estas disciplinas es la denominada *Computación Evolutiva* [12] que conjunta a su vez a la *Programación Genética*, a los *Algoritmos Genéticos* y a las *Estrategias Evolutivas*, que durante años han representado la viabilidad de utilizar este tipo de recursos para solucionar satisfactoriamente problemas de optimización a un coste computacional razonable.

Fogel explica en [11] que los tres paradigmas antes mencionados tienen diferencias entre sí, específicamente en la importancia que cobran los operadores genéticos de cruza y mutación, sin embargo, comparten funciones comunes como la reproducción, la variación aleatoria, el ímpetu para simular la evolución, la competencia y la selección, además de ser algoritmos de naturaleza estocástica.

Otra disciplina que surge como una respuesta al análisis del comportamiento social y cultural de un sistema visto como una colectividad o colonia, es la denominada *Inteligencia de Enjambre* [14] que básicamente trata de explicar

cómo se relacionan e intercambian información entre sí los individuos de estas sociedades, con la premisa de resolver una tarea común.

La integración y la coordinación de tareas de los individuos no requiere de ningún supervisor, es decir, los individuos resuelven los problemas ligados a su supervivencia de una manera distribuida y paralela. En la optimización con este tipo de técnicas, la población demanda una memoria, refiriéndose al hecho de que el proceso de mejora se dirige y encauza influenciado por la historia grupal, por la memoria de cada individuo y por el estado presente en el que cada uno se encuentra.

Aunque la inteligencia de enjambre es aplicable al comportamiento humano, se ha estudiado más a fondo sobre insectos. Los paradigmas más conocidos son la *Colonia de Hormigas* [15] y el *Cúmulo de Partículas* [16], este último inspirado en la coreografía de las parvadas de pájaros o los bancos de peces. Existen algunos otros de reciente creación y estudio, como la *Comunicación entre Luciérnagas* [17]. En la figura 1.3 se ilustra el paradigma de la Colonia de Hormigas, en donde se emula el comportamiento de las hormigas para encontrar el camino o ruta más corta entre el alimento y su nido, apoyándose de la concentración de feromonas para guiar la búsqueda.

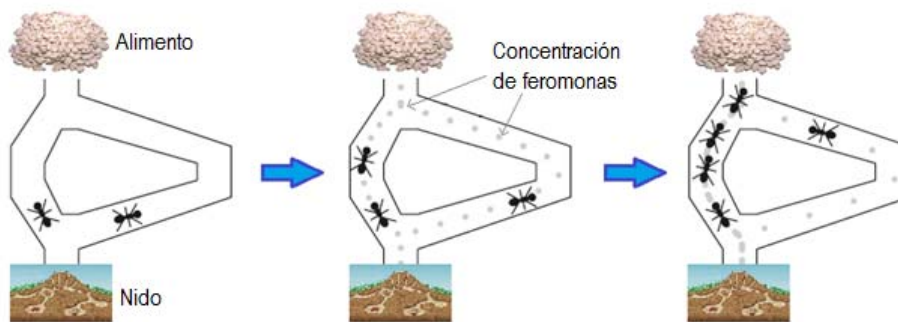


Figura 1.3. Paradigma de la colonia de hormigas.

Dentro de la literatura del área, se pueden encontrar otros modelos bioinspirados que no es sencillo clasificar, por ejemplo, la *Molécula de ADN* que se fundamenta de manera más formal en la llamada *Computación Molecular* y que básicamente ordena secuencias de cadenas de ADN para posteriormente compararlas buscando la hibridación de las mismas cuando sean complementarias [18].

Otro ejemplo son las *Redes Neuronales* que representan un modelo de aprendizaje y procesamiento automático que está inspirado en la forma en que funciona el sistema nervioso de los animales [19]. En la figura 1.4 se muestra una *red neuronal artificial (ANN)* totalmente conectada; en ésta se observa la interconexión de neuronas (elementos básicos de procesamiento) que colaboran para producir n estímulos de salida, emulando el razonamiento humano.

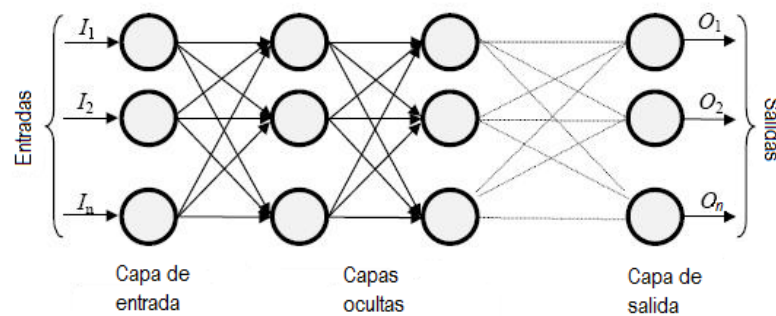


Figura 1.4. Red neuronal artificial.

Para algunos autores como Dasgupta [20], los sistemas de procesamiento de información inspirados biológicamente pueden clasificarse de la siguiente manera: sistema nervioso cerebral (redes neuronales artificiales), sistemas genéticos (algoritmos evolutivos) y sistemas inmunes (sistema inmune artificial).

1.2 Hardware evolutivo

Cuando se trasladan las técnicas evolutivas y bioinspiradas a alguna plataforma en Hardware, surge el término de *Hardware Evolutivo* [1, 3]. Es posible percibir dos ramas principales, concernientes a la operatividad del hardware, la primera de ellas es la reconfiguración automática que persigue una adaptación autónoma del hardware y, por otra parte, la implantación física de algoritmos evolutivos para acelerar algún proceso de búsqueda [2].

En la reconfiguración automática se pretende que un dispositivo electrónico pueda recibir datos del exterior, y de manera autónoma, pueda éste llevar a cabo una recombinación de su estructura interna para adaptarse al medio y mejorar su desempeño.

La implantación de algoritmos, que es la aplicación que se aborda en este trabajo de tesis, surge como una alternativa para acelerar los procesos internos y validar acciones que el hardware debe realizar, mejorando considerablemente las soluciones realizadas en software y procesadas a través de una computadora con secuenciación de instrucciones. Las opciones de implantación conllevan aproximaciones electrónicas digitales y analógicas, y en algunos casos, los menos, híbridas que combinan ambas vertientes.

El hardware evolutivo se puede implementar de dos maneras sobre silicio [35]: de forma *extrínseca*, con actualización fuera de línea, es decir, se simula todo el algoritmo y posteriormente se busca la manera de implantarlo en la plataforma hardware, o bien, de modo *intrínseco* en donde es posible integrar el algoritmo de forma embebida en el mismo dispositivo, lo cual repercute en una mayor velocidad de procesamiento y se puede hablar de reconfiguración y adaptación al medio de manera autónoma, aunque aún no se han logrado resultados totalmente independientes. Para algunos autores, es posible concebir estructuras combinadas [3].

Los dispositivos de lógica programable, con arquitecturas flexibles y robustas, como por ejemplo los *FPGAs* (*Arreglos de Compuertas Programables en Campo*) han sido durante años los recursos más utilizados para probar el

funcionamiento de toda clase de algoritmos evolutivos y bioinspirados, en un circuito integrado digital [3, 29, 35]. Lo anterior se debe a la disposición física de su arquitectura, que de manera natural está construida como un arreglo de bloques que pueden interconectarse de manera creciente entre sí para generar circuitos lógicos más complejos, aunado al gran número de pines de propósito general. En la figura 1.5 se muestra la arquitectura básica de un FPGA con sus tres elementos fundamentales: los *bloques de E/S* que propiamente se encargan del intercambio de datos con el exterior, los *bloques lógicos* (también conocidos como módulos lógicos configurables) que son las unidades en donde se implanta la lógica de solución y las *interconexiones* que son controladas por el tipo de tecnología programable que a la vez facilita la conexión entre los bloques lógicos. Entre las tecnologías programables más importantes se tienen las tecnologías SRAM y anti-fusible. La primera es reprogramable y la segunda es programable en una sola ocasión. Es claro que para prototipos experimentales es más socorrido el uso de la tecnología SRAM.

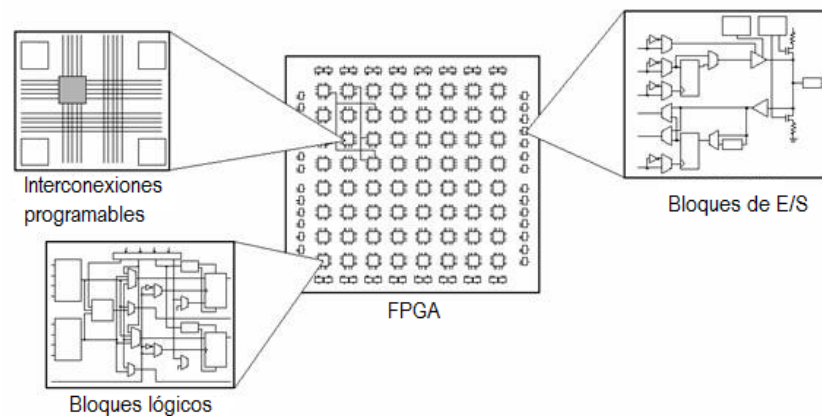


Figura 1.5. Disposición interna de un FPGA.

Las aplicaciones relativas al hardware evolutivo vertido en dispositivos FPGA son muy variadas, encontrándose desarrollos en robótica [36], en el diseño de filtros digitales [37, 38], y en el reconocimiento de patrones [39], entre otras tantas referidas principalmente a la ingeniería. La implementación analógica se realiza a nivel sustrato, proponiendo circuitos en base a transistores, resistencias y capacitores, en donde se busca evolucionar estos componentes para diseñar circuitos tolerantes a fallas y autoadaptativos [40].

1.3 Objetivo General

Diseñar un algoritmo basado en el sistema inmune artificial, con un tamaño de población reducido, para solucionar problemas de optimización numérica.

1.3.1 Objetivos Particulares

1. Realizar un estudio de los diferentes micro - algoritmos evolutivos y bioinspirados existentes en la literatura especializada, para encontrar las características que permitan diseñar un sistema inmune artificial con población reducida.
2. De los procesos biológicos del sistema inmune, se dirigirá el estudio al principio de selección clonal, considerando sus principales características: clonación y mutación.
3. Diseñar los operadores de mutación que permitan el funcionamiento correcto del algoritmo.
4. Realizar los análisis y pruebas pertinentes que permitan validar el desempeño del algoritmo.
5. Dirigir los resultados hacia arquitecturas hardware y plantear posibles aplicaciones futuras.

1.4 Justificación

Existe un interés creciente por utilizar algoritmos evolutivos y bioinspirados para optimizar procesos. La adecuación de las soluciones antes exploradas en software y ahora trasladadas a hardware, deja entrever la inmediata ventaja de la velocidad de procesamiento. Considerando las herramientas actuales en el diseño electrónico, tanto analógico como digital, es posible realizar diversos estudios y abstracciones de las técnicas conocidas.

El paralelismo inherente de los algoritmos ha sido ampliamente abordado en el diseño de *Hardware Evolutivo*, especialmente cuando se utiliza un dispositivo lógico programable en donde ha quedado de manifiesto que las velocidades de procesamiento y de convergencia son muy aceptables y no se escatima en la cantidad de recursos dentro de la arquitectura [1],[2],[3]; sin embargo, son muy pocos los estudios que pretenden reducir la arquitectura en función de los módulos lógicos, o bien, que involucren en estas tendencias a los dispositivos comerciales con restricciones de memoria y con ejecución procedural, como sucede con los microcontroladores comerciales.

La hipótesis planteada en esta tesis es que al disminuir la cantidad de individuos en la población de trabajo de un algoritmo bioinspirado, bajo algún criterio que considere los parámetros sensibles involucrados, también se reduzca el número de evaluaciones de la función objetivo, permitiendo un uso más eficiente de la memoria de datos, sin afectar de manera importante el funcionamiento del algoritmo. Con base a esta suposición, la literatura especializada reporta algunos mecanismos que buscan reducir la cantidad de evaluaciones de la función objetivo caracterizando alguna variante en el manejo de la población, como sucede con el micro-algoritmo genético [6] que representa nuestra principal referencia y que como se indicará posteriormente utiliza una convergencia nominal, un proceso de reinicialización, considera elitismo y genera ruido estocástico al completar una nueva población con nuevos individuos obtenidos aleatoriamente.

Con respecto al sistema inmune artificial para optimización, no existe referencia alguna de algoritmos con población reducida y en consecuencia tampoco se tienen evidencias de implementaciones en hardware, por lo que las aportaciones de esta tesis serán las primeras sobre el tema.

1.5 Contribuciones de la tesis

Se diseñó e implementó en software, un micro-SIA con una población de sólo 5 individuos (anticuerpos), para resolver problemas de optimización numérica global sin restricciones y con restricciones. Cabe mencionar que no existe en la literatura especializada ningún trabajo previo que aborde al SIA y que utilice una población reducida en tamaño. Para ambos casos se presentan los criterios que permitieron el funcionamiento correcto del algoritmo. Se pudo demostrar que el algoritmo propuesto presenta una convergencia más rápida en comparación a la versión estándar o extendida del SIA denominada CLONALG, que al igual que el micro-SIA propuesto, está inspirado en el principio de la selección clonal.

Debido a que se logró la meta del diseño del micro-SIA y funcionó correctamente en software con base en los experimentos realizados, se procedió a la segunda etapa de la investigación doctoral que fue la implantación del micro-SIA en una plataforma de hardware y que éste se ejecutara de forma intrínseca. El carácter intrínseco está definido por la particularidad de que el algoritmo está embebido en el dispositivo y se ejecuta de forma autónoma y no sólo recibe datos provenientes de una PC que previamente ejecuta el algoritmo y entrega resultados para configurar el hardware, como sucedería con una implementación extrínseca.

Es importante mencionar que para las aproximaciones en hardware fue necesario adecuar el micro-SIA para trabajar con una representación numérica binaria y no real, como sucedía con las versiones en software. Se realizaron dos aproximaciones para resolver el problema de la maximización de unos en una cadena finita (*maxone problem*); la primera se realizó con éxito en un

microcontrolador comercial, con recursos internos limitados, utilizando registros de 8 bits (que a la vez es el tamaño máximo de la cadena) y la segunda se programó en un dispositivo con arquitectura flexible, en este caso en un FPGA, definiendo registros de 16 bits. En este último se logró una arquitectura simplificada y modular.

El problema de la maximización de unos es una aplicación clásica en el área de reconocimiento de patrones; no obstante resulta interesante su implementación en hardware, no siendo una tarea trivial debido a las características modulares de un algoritmo bioinspirado, es decir, la realización física de módulos que generen números aleatorios, ordenen datos y apliquen los respectivos operadores genéticos, además se requiere de un bloque adicional que permita el conteo eficiente de los bits puestos a uno para medir la aptitud de un individuo. Finalmente, se presenta una aplicación de reconocimiento de caracteres que hace uso del micro-SIA, implementado en el FPGA, para obtener el caracter más parecido a otro corrupto que se utiliza como entrada en el sistema de reconocimiento, utilizando una matriz de 35 bits para cada caracter (7 filas y 5 columnas).

1.6 Organización de la tesis

El presente documento está organizado en 5 capítulos. En el Capítulo 1 se presenta la introducción al trabajo realizado. En este mismo capítulo se hace mención de los objetivos general y particulares de la tesis y se expone la justificación de la investigación. Al final de este capítulo se indican cuáles son las contribuciones de la tesis.

El segundo capítulo del documento, Capítulo 2, versa en las generalidades del sistema inmune artificial y se presenta el estado del arte sobre micro-algoritmos.

El Capítulo 3 se explica la propuesta presentada y cómo funciona. Se hace hincapié en la manera en la que se concibió el micro-algoritmo, denominado micro-SIA, para resolver problemas de optimización numérica, y cuáles son sus características. En este mismo apartado se incluyen y comentan las pruebas y resultados experimentales de la realización en software.

En el Capítulo 4 se presentan las arquitecturas en hardware que se diseñaron e implementaron para el micro-SIA, con sus respectivos experimentos.

El último capítulo, Capítulo 5, se presentan las conclusiones derivadas de la investigación doctoral y se detallan las propuestas de trabajos a futuro que surgen a partir del desarrollo de esta tesis.

Finalmente se incluyen las referencias estudiadas y se presentan 4 anexos. El Anexo A contiene el listado de las funciones de prueba utilizadas para los experimentos del Capítulo 3, correspondientes a la optimización numérica sin restricciones. El contenido del Anexo B es el listado de las funciones de prueba utilizadas en la optimización numérica con manejo de restricciones, de los experimentos realizados en el mismo Capítulo 3.

Los anexos C y D sirven para mostrar dos trabajos que surgieron de manera paralela al desarrollo de esta tesis y que fortalecieron el conocimiento adquirido durante la investigación. El Anexo C presenta el diseño de un micro-algoritmo de Evolución Diferencial, siguiendo la misma metodología que se utilizó para el diseño del micro-SIA.

Finalmente, en el Anexo D se incluye el diseño de una arquitectura bioinspirada para una unidad de modulación por ancho de pulso (PWM), que se utiliza como una alternativa para la conversión de un dato digital a su contraparte analógica. Esta arquitectura fue abstraída de la hibridación de cadenas de ADN.