



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
MAESTRÍA EN CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

**ALGORITMOS GENÉTICOS VS ENJAMBRES DE PARTÍCULAS:
BÚSQUEDA DE PARÁMETROS EN UNA DINÁMICA COMPLEJA**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS (COMPUTACIÓN)

PRESENTA:
ANDRÉS ALDANA GONZÁLEZ

TUTORES:
DR. PEDRO EDUARDO MIRAMONTES VIDAL
FACULTAD DE CIENCIAS
DR. GUSTAVO CARLOS MARTÍNEZ MEKLER
INSTITUTO DE CIENCIAS FÍSICAS

MÉXICO, D. F. AGOSTO 2015



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*Para mi esposa Andrea,
por acompañarme en cada paso*

*Para mi familia,
por apoyarme siempre*

Agradecimientos

La elaboración de esta tesis ha sido posible gracias no solamente al apoyo y contribución de muchas personas que han participado de forma directa en su desarrollo, sino también a aquellos que aportaron en mi formación y que de alguna forma me dieron los elementos necesarios para realizarla.

Quiero iniciar agradeciendo profundamente a mis tutores, el Dr. Pedro Eduardo Miramontes Vidal y el Dr. Gustavo Carlos Martínez Mekler quienes siempre estuvieron dispuestos a compartir su tiempo, conocimiento y experiencia conmigo. Gracias a ellos esta tesis ha sido posible pues han sido guías integrales en la elaboración de este trabajo.

Quiero agradecer también a mi hermano, el Dr. Maximino Aldana González, pues ha sido siempre una fuente de inspiración, motivación y admiración y fue él quien me invitó a participar en el proyecto que motivó esta tesis. Son invaluable sus consejos y enseñanzas a lo largo de mi vida y estaré siempre agradecido con él por haberme inculcado el amor a la ciencia.

Agradezco mucho a mis sinodales, la Dra. Katya Rodriguez Vazquez, el Dr. Carlos Gershenson García y el Dr. Alberto Darszon Israel cuyos comentarios y correcciones no solamente mejoraron notablemente la calidad de este trabajo, sino también me permitieron incrementar mi conocimiento sobre temas relacionados con esta tesis.

Este trabajo se ha nutrido enormemente de los comentarios de diversas personas. En ese sentido, estoy infinitamente agradecido con mi esposa la matemática Andrea Falcón, quien ha sido una fuente constante de respeto, confianza y amor, por todas las veces que atentamente me escuchó hablar de los temas abordados en cada paso de la realización de esta tesis. Sus comentarios, recomendaciones y observaciones han sido completamente enriquecedores y formativos y su participación ha sido sin duda uno de los pilares en mi formación durante la maestría.

Quiero agradecer a mi mamá, la Sra. Catalina González Espinosa quien siempre me ha querido y apoyado de forma incondicional. Sin duda sin su amor, comprensión, paciencia, respeto y confianza esta tesis no habría sido realizable. Es gracias a su esfuerzo y dedicación que he podido hacer una carrera científica.

También agradezco a mi hermana, la C. (casi) Maestra en relaciones internacionales Claudia Matilde Aldana González que a lo largo de mi vida ha participado de

AGRADECIMIENTOS

manera activa en mi formación no solamente como estudiante sino como persona e individuo de esta sociedad. Sus enseñanzas y consejos me han ayudado a superar momentos difíciles y me permitieron desarrollar la personalidad necesaria para superar mis estudios desde preescolar hasta el posgrado.

Agradezco especialmente al Dr. Hector Benítez Pérez y al Dr. Juan Mario Peña Cabrera por el apoyo y la flexibilidad que me han brindado en la elaboración de esta tesis, pues la confianza y facilidades que me han proporcionado no solamente me permitieron terminarla, sino también impulsar mi conocimiento y aprendizaje en muy diversos temas.

Deseo agradecer a los profesores y compañeros de la maestría y al Posgrado en Ciencias e Ingeniería de la Computación de la UNAM por haberme permitido adquirir el conocimiento y experiencia que sin duda han marcado mi formación y por ende mi futuro.

Quiero agradecer a CONACyT por el financiamiento que me permitió dedicarme tiempo completo a la maestría. Sin duda, el apoyo a la educación es la mejor inversión en el progreso del país y CONACyT ha cumplido exitosamente con esa labor. Así mismo, quisiera agradecer a PAPIIT ya que con su apoyo financiero fue posible comprar el equipo necesario para desarrollar la investigación que en este trabajo se plasma.

Finalmente quiero agradecer a todos aquellos familiares y amigos que en algún momento han emitido observaciones y comentarios que han aportado en mi formación académica y personal.

A todos ustedes: ¡¡Les estoy profundamente agradecido!!

Andrés Aldana González

Índice general

Agradecimientos	III
Introducción	3
1. Antecedentes	5
1.1. Búsqueda y optimización	5
1.2. Algoritmos genéticos	6
1.2.1. El algoritmo genético simple	7
1.2.2. Selección	8
1.2.3. Codificación	9
1.2.4. Cruza	10
1.2.5. Mutación	11
1.3. Optimización por enjambres de partículas (PSO)	12
1.3.1. El algoritmo base	13
1.3.2. Vecindades	14
1.3.3. Inercia	15
1.4. Fecundación y redes de regulación bioquímica	16
1.4.1. La red del calcio	17
1.4.2. Dinámica discreta	19
1.4.3. Dinámica de Glass	20
2. Algoritmos genéticos vs optimización por enjambres de partículas	23
2.1. Discusión previa	24
2.2. Metodología de comparación	27
3. Caso de estudio “La red del calcio”	33
3.1. Contexto	33
3.2. Transformación de los datos	36
3.3. Generando las señales faltantes	37
3.4. Búsqueda de parámetros para la red del calcio	38
3.4.1. La función de adaptación	39
3.4.2. Ejecución	40
3.4.3. Resultados	42
4. Conclusiones	49
4.1. Contribuciones	50
4.2. Trabajo futuro	50

Bibliografía

53

Introducción

Desde los inicios de la computación, especialistas de diversas áreas como ingeniería, ciencias de la computación, biología, matemáticas y física abordaron de manera recurrente el problema de la optimización y búsqueda de parámetros en espacios multidimensionales. En un problema de optimización con un único objetivo, cada punto en el espacio de búsqueda tiene asociado un valor numérico llamado *valor de adaptación* por analogía con la adaptabilidad evolutiva en la naturaleza, que indica qué tan bueno o malo es este punto como solución al problema. Así, el conjunto de los valores de adaptación forma una “superficie” multidimensional llamada *función de adaptación*, que en general puede presentar geometrías complicadas (irregulares, con múltiples máximos y mínimos locales, discontinuas, no diferenciables, con regiones planas, mínimos o máximos de aguja o combinaciones de las anteriores). Por otra parte, es frecuente encontrar espacios de búsqueda con una gran cantidad de posibles soluciones. La optimización consiste en encontrar aquellos valores en el espacio de búsqueda que representan la mejor solución (o una solución cercana a la mejor) para un problema dado. En general, la mejor solución es aquel punto (o puntos) del espacio de búsqueda que tiene asociado el valor de adaptación máximo (o mínimo¹). Así, la optimización generalmente es un proceso complicado debido a la geometría de la función de adaptación y al tamaño del espacio de búsqueda.

En este sentido, a lo largo del tiempo fueron desarrollados diversos algoritmos que intentan encontrar las mejores soluciones en un espacio de búsqueda. Rápidamente los desarrolladores de dichos algoritmos se dieron cuenta de que los algoritmos deterministas que intentan encontrar la mejor solución no son factibles en espacios de búsqueda con funciones de adaptación geoméricamente complicadas debido a la estructura geométrica de la función o al inmenso tiempo de ejecución necesario para asegurar la mejor solución. Así, la restricción de “obtener la solución óptima” es sustituida frecuentemente por otra menos estricta: “obtener una solución cercana a la óptima”.

Para explorar eficientemente el espacio de búsqueda, diversos algoritmos heurísticos fueron desarrollados para obtener soluciones aceptables. Dos de estos algoritmos son los ampliamente citados algoritmos genéticos y la más reciente optimización por enjambre de partículas (PSO por sus siglas en inglés). Ambas técnicas han demostrado ser efectivas y su aplicación abarca problemas en muy diversas áreas.

¹Cuando la optimización consiste en encontrar el mínimo de la función

En esta tesis hacemos una comparación numérica entre un tipo de algoritmo genético (algoritmos genéticos de codificación real) y optimización por enjambres de partículas utilizando como caso de estudio un modelo de redes de reacciones bioquímicas ligadas a la motilidad de los espermatozoides del erizo de mar que intenta explicar la fecundación externa en esta especie, en el cual el objetivo es encontrar aquellos parámetros del modelo que le permiten aproximar lo mejor posible un conjunto de mediciones experimentales. Analizamos las similitudes y diferencias entre dichas técnicas de optimización y proporcionamos información sobre su eficiencia y eficacia. Esta tesis no pretende calificar a una técnica como *mejor* en general que la otra, sino dar elementos que permitan decidir cuál implementación utilizar ante un problema de características similares al caso de estudio planteado. Por otra parte, la metodología de comparación propuesta puede utilizarse en diversos algoritmos de búsqueda y optimización basados en agentes. Paralelamente el análisis en el caso de estudio mencionado nos ayuda a comprender mejor la forma en que los espermatozoides se aproximan a los gametos femeninos para ser fecundados, lo cual es uno de los problemas principales de la biología y tiene implicaciones en la fecundación y fertilidad en diversas especies.

La primera parte de esta tesis contenida en el capítulo 1 provee al lector con los elementos teóricos necesarios para comprender a los algoritmos genéticos, los enjambres de partículas y el contexto en el cuál ambos son aplicados en el caso de estudio mencionado. En el capítulo 2 se muestran diversas investigaciones donde se hicieron comparaciones similares y se estudian algunas técnicas de comparación fundamentales que utilizamos en esta tesis, se presenta también un análisis de las diferencias y similitudes entre algoritmos genéticos y enjambres de partículas y finalmente se describe el diseño del análisis numérico entre ambas técnicas de optimización. El capítulo 3 contiene la comparación numérica realizada entre algoritmos genéticos y optimización por enjambres de partículas en el caso de la red de señalización bioquímica antes mencionada y se proveen resultados que proporcionan información sobre el comportamiento de ambos algoritmos en este problema. Finalmente, el capítulo 4 presenta las conclusiones de esta tesis.

Capítulo 1

Antecedentes

1.1. Búsqueda y optimización

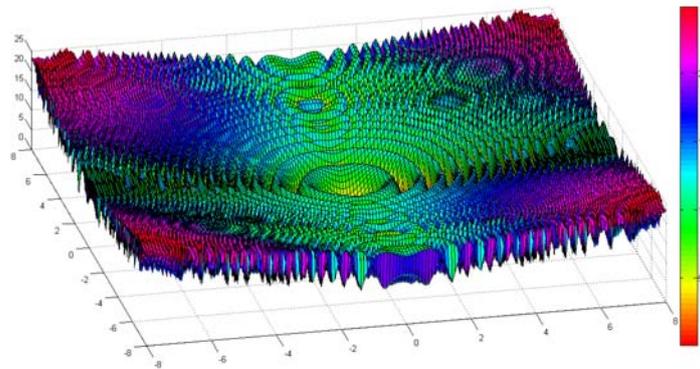


Figura 1.1: Una función geoméricamente complicada con discontinuidades, puntos no diferenciables y diversos mínimos y máximos locales

Con frecuencia, las personas dedicadas a ciencia e ingeniería se encuentran con la necesidad de hacer elecciones entre distintas soluciones posibles de un determinado problema para obtener resultados relevantes. La optimización es el proceso de tomar estas elecciones de la mejor manera. La noción de “resultados relevantes” se refiere a encontrar las mejores soluciones del problema. Así, en un problema de optimización las soluciones candidatas son comparadas y contrastadas, lo cual indica que la calidad de las soluciones es de gran importancia [1]. En este sentido generalmente la mejor solución está dada por el punto en el espacio de búsqueda que tiene asociado el valor de adaptación más alto, pero existen trabajos donde el valor de adaptación puede visualizarse como un valor de *error* asociado a cada punto del espacio. En estos casos, la mejor solución está dada por el punto del espacio que tiene asociado el mínimo error y por ende el valor de adaptación mínimo [1, 2].

Generalmente, la solución de un problema de optimización, dado por punto en el

espacio de búsqueda con el valor de adaptación mínimo o máximo no es fácil de encontrar debido a que se encuentra en un gran espacio de búsqueda o a que la función de adaptación asociada a este espacio es geoméricamente complicada como la de la figura 1.1. En la práctica, este tipo de problemas son difíciles de resolver debido a la imposibilidad de probar y comparar cada una de las posibles soluciones para encontrar la solución óptima dentro de un tiempo aceptable. En problemas cuyo espacio de soluciones es pequeño, es posible examinar cada una de las posibles soluciones y encontrar la óptima entre ellas. Sin embargo, en problemas con un extenso espacio de soluciones el uso de algoritmos deterministas como fuerza bruta o simplex se vuelve impráctico y es imposible garantizar la solución óptima en un tiempo razonable. En este tipo de problemas es común encontrar que el espacio de búsqueda no solamente es muy extenso sino discontinuo o ruidoso, por lo cual encontrar la solución óptima es aún más difícil. Finalmente, en la mayoría de estos problemas no se requiere encontrar la solución global absoluta, sino alguna solución o conjunto de soluciones que correspondan a un valor inferior a alguna función objetivo. Esto es, que la solución o soluciones encontradas sean suficientemente buenas [1].

Debido a ello, en los últimos años se enfatizó el uso de técnicas de búsqueda aproximada para resolver problemas de optimización porque pueden encontrar soluciones cercanas a la óptima dentro de un tiempo de cómputo aceptable. Aunque se desarrollaron diversas y muy variadas técnicas de búsqueda aproximada, en esta tesis abordamos dos de las más utilizadas en los últimos años: algoritmos genéticos y optimización por enjambres de partículas [1].

1.2. Algoritmos genéticos

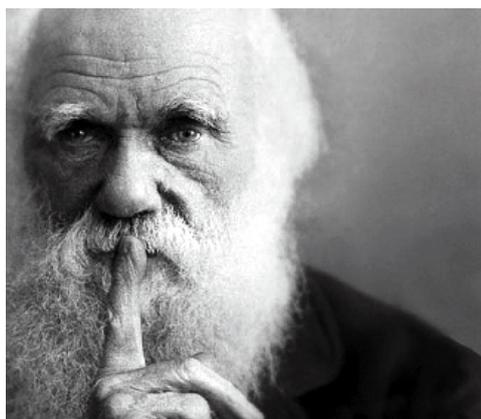


Figura 1.2: Charles Darwin postuló su teoría evolutiva en “el origen de las especies” en 1859. Los algoritmos genéticos se inspiran en la evolución biológica darwiniana y su base molecular para optimizar funciones.

1.2. ALGORITMOS GENÉTICOS

No existe una definición rigurosa de “algoritmo genético” aceptada universalmente por la comunidad de la computación evolutiva que diferencie a los algoritmos genéticos de otras estrategias evolutivas. Sin embargo, podemos referirnos a los algoritmos genéticos como técnicas de búsqueda y optimización que intentan reproducir los principios evolutivos y procesos cromosómicos de la genética. Un algoritmo genético inicia con un conjunto aleatorio de posibles soluciones, usualmente codificadas en cadenas de componentes binarios (aunque en esta tesis, veremos un tipo diferente de codificación). A cada posible solución se le asigna un valor de adaptación que está directamente relacionado a la calidad de ésta en el problema de optimización. La población de posibles soluciones es modificada y transformada en una nueva población a través de tres operadores genéticos: selección, cruce y mutación. En un algoritmo genético, estos tres operadores son aplicados de manera iterativa produciendo nuevas generaciones de soluciones posibles hasta que se cumple un criterio de paro [1–3].

La mecánica de los algoritmos genéticos toma su motivación de la evolución darwiniana en la naturaleza, la cual sigue dos reglas simples:

1. Si por medio de un proceso genético se crea una descendencia con un nivel de adaptación por encima del promedio, los individuos de esta descendencia tienden a sobrevivir más tiempo que los individuos promedio y en consecuencia tienen más oportunidades para producir descendientes que hereden sus características [1].
2. Si por el otro lado, una descendencia es creada con un valor de adaptación por debajo del promedio, usualmente ésta no sobrevive y es eliminada de la población [1].

El principio de enfatizar buenas soluciones y eliminar malas soluciones es una característica fundamental en un algoritmo de optimización [1].

Los algoritmos genéticos no son tan complejos como los procesos genéticos en la naturaleza, sino abstracciones simples de los procesos evolutivos naturales. Sin embargo, son robustos, eficaces y mediante ellos se pudieron resolver diversos problemas de búsqueda y optimización en diversas áreas [1–3].

1.2.1. El algoritmo genético simple

La versión más sencilla de algoritmo genético funciona como sigue [2, 3]:

1. Se inicia con una población de n individuos generada aleatoriamente en el espacio de posibles soluciones. Cada individuo está representado por un cromosoma (cadena) de m bits (genes). A esta construcción se le llama “solución candidata”.
2. Se calcula el valor de adaptación $f(x)$ para cada cromosoma x en la población.
3. Se repite el siguiente procedimiento hasta que se hayan creado n descendientes:

- a) Se seleccionan un par de cromosomas de la población actual (padres), donde la probabilidad de ser seleccionado es una función creciente con el valor de adaptación (mejor adaptación significa mayor probabilidad de ser seleccionado). En el proceso de selección un cromosoma puede ser seleccionado más de una vez para ser padre (selección con reemplazo).
- b) Con probabilidad p_c (probabilidad de cruza), se cruzan los cromosomas padres en una posición seleccionada aleatoriamente para crear dos nuevos descendientes. Supongamos que los cromosomas padres son $a = [a_1, a_2, \dots, a_m]$ y $b = [b_1, b_2, \dots, b_m]$ y de forma aleatoria seleccionamos la posición j para llevar a cabo la cruza. Si esta no se lleva a cabo, entonces se generan dos copias idénticas de los cromosomas padres a y b . Si la cruza se lleva a cabo entonces se generan dos descendientes h_1 y h_2 de la siguiente forma:

$$\begin{aligned} h_1 &= [a_1, \dots, a_j, b_{j+1}, \dots, b_m] \\ h_2 &= [b_1, \dots, b_j, a_{j+1}, \dots, a_m] \end{aligned}$$

La probabilidad de cruza esta definida como la probabilidad de que dos padres se crucen en una única posición j . Existen versiones de algoritmos genéticos en los que se llevan a cabo cruza multi-posición, en las cuales la probabilidad de cruza determina el número de posiciones en los cuales se cruzan los cromosomas padres.

- c) Se realiza mutación puntual en cada bit de los descendientes con probabilidad p_m (probabilidad de mutación), y se coloca a los cromosomas resultantes en la nueva población.
- d) Si el tamaño de la población n es impar, se descarta un individuo de la nueva población de forma aleatoria.

4. Se reemplaza a la población actual con la nueva población

5. Ir al paso 2.

A cada iteración de este proceso se le llama *generación*. Generalmente, un algoritmo genético se itera durante un número de generaciones generalmente dependiente del problema a resolver. Al conjunto completo de generaciones se le llama *ejecución*. Al final de cada ejecución, generalmente hay uno o más cromosomas de la población altamente adaptados (con valores de $f(x)$ altos o bajos según sea el caso) [1–3].

Desde la aparición de los algoritmos genéticos, se han desarrollado diversos operadores de selección, cruza y mutación para mejorar el funcionamiento del algoritmo original.

1.2.2. Selección

El propósito de la selección de los cromosomas padres que generan descendientes para la nueva población es que los cromosomas con mejor valor de adaptación

1.2. ALGORITMOS GENÉTICOS

hereden algunas de sus características a la nueva generación esperando que su descendencia tenga valor de adaptación aún mejor. La selección debe ser equilibrada con variaciones que aportan la cruce y mutación, pues una selección demasiado estricta de los cromosomas con mejor valor de adaptación significaría que los individuos subóptimos reemplazarían a la población reduciendo la diversidad genética necesaria para crear nuevos cambios. Por otra parte, una selección aleatoria de los cromosomas resultaría en una evolución demasiado lenta. Debido a esto fueron desarrolladas diversas técnicas de selección, entre las cuales destacan el *método de la ruleta*, *escalado sigma*, *elitismo*, *selección de Boltzmann*, *selección por ranking* y *selección por torneo* [1, 3]. Entre ellas elegimos *selección por torneo binario* para desarrollar los algoritmos genéticos descritos en este trabajo por tres razones:

- Es un método de selección computacionalmente más eficiente que otros métodos ya que requiere una cantidad mínima de operaciones [3, 4].
- Al tomar el individuo mejor adaptado de forma aleatoria siguiendo cierta probabilidad, este algoritmo de selección equilibra la velocidad de evolución y convergencia del algoritmo genético al mantener diversidad en la población y al mismo tiempo dar preferencia de selección a las mejores soluciones [3, 4].
- La evidencia experimental desarrollada en este trabajo mostró los mejores resultados con este método.

La selección por torneo funciona de la siguiente forma: Se eligen dos cromosomas al azar entre la población. Se escoge un número aleatorio r en el rango $(0, 1)$ y un parámetro k en el mismo rango. Si $r < k$, se selecciona al cromosoma con el mejor valor de adaptación como uno de los padres. De otra forma, se selecciona como padre al cromosoma con el peor valor de adaptación. Ambos cromosomas son reingresados a la población y pueden ser elegidos nuevamente [3, 4].

1.2.3. Codificación

En el algoritmo genético clásico, el espacio de soluciones está representado por cadenas de bits, donde el conjunto de estados de cada variable en las soluciones potenciales está representado por un fragmento de esa cadena. En los algoritmos genéticos de codificación real, las variables del problema se representan directamente. Por ello, una solución está representada por un vector de números reales de la forma (x_1, x_2, \dots, x_n) donde cada variable x_i puede tomar valores en un intervalo permitido (m_i, M_i) [1, 3].

En los algoritmos genéticos de codificación binaria, pueden existir dificultades al tratar de resolver problemas con un espacio de búsqueda continuo. Por ejemplo, ya que los algoritmos genéticos con codificación binaria discretizan el espacio de búsqueda al utilizar cadenas binarias de longitud finita, la precisión arbitraria no es fácil de obtener. En espacios continuos, la granularidad del espacio de búsqueda y por ende la calidad de las soluciones está definida por la longitud de la cadena binaria, lo cual complica la obtención de buenos resultados. Además, la codificación

binaria genera *Hamming cliffs*, es decir, grandes distancias de hamming entre enteros adyacentes (por ejemplo, 0111 y 1000 representan 7 y 8, pero tienen una distancia de Hamming de 4). Aunque se realizaron diversos trabajos para eliminar Hamming cliffs en la codificación binaria a por medio de código Gray ([5, 6]), el problema de la granularidad persiste en espacios continuos. El uso directo de variables en el algoritmo genético con codificación real elimina estas dificultades [1, 3].

Un beneficio más de la codificación real es que dado que las variables del problema son inicializadas directamente, pueden ser utilizadas para calcular el valor de adaptación del cromosoma de forma más rápida. Debido a ello y dada la naturaleza continua del espacio de búsqueda en el caso de estudio tratado en el capítulo 3, decidimos utilizar codificación real para la implementación de los algoritmos genéticos utilizados en el desarrollo de esta tesis. Sin embargo, los operadores de cruce y mutación son diferentes a los utilizados en el algoritmo genético simple. En las subsecciones posteriores explicamos el comportamiento de estos operadores para el algoritmo genético con codificación real [1, 3]:

1.2.4. Cruza

Desde el algoritmo genético original, se desarrollaron diversas técnicas para este operador genético en el caso de la codificación real. La *cruza discreta simple*, *cruza discreta de dos puntos* y *cruza discreta uniforme* son similares a los operadores de cruce para el algoritmo genético de codificación binaria. Sin embargo, estos operadores tienden a mantener las características de la población inicial. Los operadores *cruza aritmética* y *cruza de media geométrica* combinan los valores de los cromosomas padres posición por posición, pero tienen el inconveniente de que se pierde diversidad genética ya que tienden a llevar a los individuos hacia la mitad del intervalo (m_i, M_i) . Finalmente, los operadores *cruza SBX (simulated binary crossover)*, *cruza BLX- α (blend crossover, alpha)* y *cruza BGA (Breeder Genetic Algorithm)* [7] utilizan distribuciones probabilísticas que permiten con mayor frecuencia crear descendientes cercanos a los padres y con poca frecuencia descendientes lejanos a los padres. Debido a ello, para esta tesis elegimos utilizar el operador de cruce BLX- α , pues permite mantener la diversidad genética entre la población [1, 8].

Suponiendo que en el proceso de selección se escogieron los padres (x_1, x_2, \dots, x_n) y (y_1, y_2, \dots, y_n) , este operador de cruce funciona de la siguiente forma [1]:

1. Se generan dos números aleatorios α, β en el intervalo $[0, 1]$.
2. Cada valor h_i en el nuevo descendiente se obtiene con la fórmula:

$$h_i = r_i + \beta|r_i - s_i|$$

donde:

$$\begin{aligned} r_i &= \min(x_i, y_i) - \alpha|x_i - y_i| \\ s_i &= \max(x_i, y_i) + \alpha|x_i - y_i| \end{aligned}$$

1.2. ALGORITMOS GENÉTICOS

El valor α genera valores para el hijo dentro del intervalo $(\min(x_i, y_i) - \alpha|x_i - y_i|, \max(x_i, y_i) + \alpha|x_i - y_i|)$ [8].

1.2.5. Mutación

Al igual que en el operador de cruza, se implementaron diversas técnicas de mutación para los algoritmos genéticos de codificación real. Por ejemplo la *mutación uniforme* en la que el gen mutado es reemplazado por un número aleatorio con distribución uniforme en el rango de los padres [1]. Sin embargo, es deseable que las mutaciones se den dentro de un intervalo alrededor del gen original. La *mutación BGA (Breeder Genetic Algorithm)* [7] y la *mutación no uniforme* [1] presentan esta característica y permiten diversidad genética en la población a la vez que preservan las características de las mejores soluciones [1]. En esta tesis utilizamos *mutación no uniforme* pues requiere menos parámetros que la *mutación BGA*. Para implementar el operador de mutación, el valor del gen que debe ser mutado es reemplazado por un valor vecino utilizando una distribución de probabilidad polinomial con media en el valor actual y cuya varianza es una función de un determinado índice de distribución η_m . Así para realizar una mutación se define un factor de perturbación δ como sigue [1]:

$$\delta = \frac{c - p}{\Delta_{max}}$$

donde Δ_{max} es una cantidad fija que representa la máxima perturbación permitida en el valor original del gen p y c es el valor mutado. Este es calculado con una distribución de probabilidad que depende del factor de perturbación δ :

$$p(\delta) = 0,5(\eta_m + 1)(1 - |\delta|)^{\eta_m} \quad (1.1)$$

Esta distribución de probabilidad es válida en el rango $(-1, 1)$. Así, para crear un valor mutado se genera un número aleatorio u en el rango $(0, 1)$ y se utiliza la siguiente ecuación para crear el factor de perturbación $\bar{\delta}$ correspondiente a u usando la distribución de probabilidad anterior:

$$\bar{\delta} = \begin{cases} (2u)^{\frac{1}{\eta_m+1}} - 1 & \text{si } u \leq 0,5 \\ 1 - [2(1 - u)]^{\frac{1}{\eta_m+1}} & \text{si } u > 0,5 \end{cases} \quad (1.2)$$

Finalmente, el valor mutado c generado a partir del gen p se calcula con la siguiente ecuación [1]:

$$c = p + \bar{\delta}\Delta_{max} \quad (1.3)$$

La codificación real definida con los operadores genéticos cruza BLX- α , mutación no uniforme y selección por torneo permite a los algoritmos genéticos explorar espacios continuos de manera eficiente con un grado de precisión arbitraria, sin problemas con la distancia de Hamming entre posibles soluciones y manteniendo diversidad genética a la vez que se explora el espacio de soluciones cercanas a las subóptimas. Debido a ello, en esta tesis utilizamos algoritmos genéticos de codificación real con los operadores descritos anteriormente.

Desde su aparición, se utilizaron algoritmos genéticos exitosamente en diversas áreas de las ciencias y las ingenierías y proporcionaron una herramienta poderosa y eficaz para explorar espacios de búsqueda geoméricamente complicados e irregulares. Una técnica popular y eficaz que también utiliza analogías provistas por la naturaleza para explorar espacios de este tipo es la denominada optimización por enjambres de partículas.

1.3. Optimización por enjambres de partículas (PSO)



Figura 1.3: Imagen de una bandada de pájaros estorninos sobrevolando una estación de tren en Roma

Al igual que los algoritmos genéticos, la metodología de enjambres de partículas es un poderoso mecanismo para optimizar funciones no lineales propuesto por Kennedy y Eberhart en [9]. La intención inicial estaba ubicada en la inteligencia artificial y explotaba analogías simples de la interacción social en vez de habilidades cognitivas individuales. Las primeras simulaciones intentaban ser una analogía a las bandadas de pájaros buscando maíz. Estas simulaciones se convirtieron rápidamente en un poderoso mecanismo de optimización llamado *optimización por enjambres de partículas* o en inglés *Particle Swarm Optimization*, mejor conocido como *PSO* [10].

En PSO un número de agentes simples (las partículas) se posiciona de forma aleatoria en el espacio de búsqueda de algún problema o función. Esto es, el sistema es inicializado con una población de soluciones potenciales aleatorias. Cada partícula evalúa su valor de adaptación respecto a la posición en la que se encuentra en el espacio. Las partículas determinan su movimiento a través del espacio de búsqueda combinando algunos aspectos de su posición actual, la mejor posición visitada (mejor valor de adaptación local) y la mejor posición que visitada por otros miembros del enjambre (mejor valor de adaptación global), junto con perturbaciones aleatorias. La siguiente iteración se lleva a cabo una vez que todas las partículas actualizaron su posición. Eventualmente, el enjambre completo se mueve cerca de un óptimo de

1.3. OPTIMIZACIÓN POR ENJAMBRES DE PARTÍCULAS (PSO)

la función de adaptación como una bandada de pájaros que colectivamente intentan encontrar comida [9–11].

Cada partícula recuerda las coordenadas de la mejor solución que visitó hasta el momento junto con su valor de adaptación asociado. A este valor se le llama $pbest$. En la versión “Global” de particle swarm optimization, las partículas llevan registro de las coordenadas de la mejor solución visitada por el enjambre, junto con su valor de adaptación. A este valor se le llama $gbest$ [9–11].

En cada paso de tiempo, se modifica la velocidad de cada partícula acelerándola hacia sus coordenadas $pbest$ y $gbest$. Esta aceleración es afectada por una perturbación aleatoria que consiste en dos números aleatorios generados de forma independiente para los valores $pbest$ y $gbest$ [9–11].

En [11] y [10] se reportan mejores resultados al utilizar una versión “local” de este mecanismo en la que además de $pbest$, cada partícula lleva registro de la mejor solución, que llamamos $lbest$, obtenida dentro de una vecindad topológica *local* construida para esa partícula. De esta forma, el enjambre tiene menos posibilidades de caer en mínimos locales. Debido a ello, utilizamos la versión local de PSO para esta tesis.

Finalmente, Shi y Eberhart [12] (1998) utilizan el concepto de inercia para controlar la exploración y explotación del espacio de búsqueda a través de las iteraciones del enjambre. Este concepto es utilizado para controlar la velocidad de las partículas con el objetivo de explorar el espacio de manera más eficiente [13, 14]. Por ello, agregamos este concepto a la implementación de PSO utilizada en esta tesis.

1.3.1. El algoritmo base

Supóngase un espacio de búsqueda de dimensión n . Cada partícula i del enjambre tiene un vector de posición $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ y un vector de velocidad $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$. A través del espacio de búsqueda, las partículas recuerdan la mejor posición ($pbest$) visitada $p_i = (p_{i1}, p_{i2}, \dots, p_{in})$. Además, todas las partículas conocen la mejor posición visitada en conjunto por el enjambre ($gbest$) $p_g = (p_{g1}, p_{g2}, \dots, p_{gn})$. Finalmente, cada partícula actualiza su vector de velocidad v_i en cada iteración tomando en cuenta los valores p_i y p_g de la siguiente forma [9–14]:

$$v_i(t+1) = v_i(t) + c_1\gamma_{1i}(p_i(t) - x_i(t)) + c_2\gamma_{2i}(p_g(t) - x_i(t)) \quad (1.4)$$

donde t representa el índice de la iteración, γ_{1i} y γ_{2i} son números aleatorios en el intervalo $[0, 1]$ que representan perturbaciones en la velocidad debido a diversos factores biológicos y c_1 y c_2 son dos constantes positivas que pueden ser vistas como el factor cognitivo personal y el factor cognitivo social respectivamente en cada partícula.

Para calcular el movimiento de cada partícula, tomamos en cuenta su posición anterior y el vector velocidad de la ecuación 1.4, de donde la posición de cada partícula

i se actualiza en cada iteración de la siguiente forma [9–14]:

$$x_i(t+1) = x_i(t) + v_i(t) \quad (1.5)$$

Sin pérdida de generalidad, podemos asumir que la optimización consiste en encontrar la posición con el valor de adaptación que representa el menor error y así, deseamos encontrar el valor mínimo de la función de adaptación. El algoritmo base de la optimización por enjambre de partículas se define como sigue [9–14]:

1. Se inicializan todas las partículas con posiciones y velocidades aleatorias.
2. Para cada partícula i en el enjambre:
 - a) Se evalúa la función de adaptación f en la posición x_i .
 - b) Se compara el valor de adaptación de la partícula con el valor de adaptación de la mejor posición visitada por la partícula p_i . Si $f(x_i) < f(p_i)$ entonces $p_i = x_i$.
 - c) Se compara el valor de adaptación de la partícula con el valor de adaptación de la mejor posición visitada por el enjambre p_g . Si $f(x_i) < f(p_g)$ entonces $p_g = x_i$.
 - d) La posición x_i de la partícula se actualiza con 1.4 y 1.5.
3. Ir al paso 2 y repetir hasta que se cumpla algún criterio.

El algoritmo anterior fue propuesto por Kennedy y Eberhart en 1995 [9]. Desde entonces se modificó para mejorar la exploración y explotación del espacio de soluciones. Una de estas modificaciones es la inclusión de vecindades para formar la versión “local” de PSO.

1.3.2. Vecindades

Basándose entre otras cosas en descubrimientos hechos en simulaciones de eventos sociales, se decidió diseñar una versión “local” del concepto de enjambre de partículas en donde cada una de ellas está conectada únicamente con un subconjunto del enjambre (vecindad). Así, en vez de tomar en cuenta la mejor posición visitada por el enjambre ($gbest$), cada partícula considera la mejor posición que visitaron los miembros dentro de su vecindad ($lbest$) [9–11].

Diversas estrategias se desarrollaron para definir la vecindad de una partícula. Por una parte, las vecindades dinámicas consideran que los miembros de la vecindad de una partícula cambian con cada iteración. Sin embargo, la mayoría de estas estrategias son computacionalmente costosas. Por otro lado, en las vecindades estáticas los miembros de la vecindad son definidos al principio del algoritmo para cada partícula y no sufren cambios durante el desarrollo del proceso de búsqueda. Comparadas con el algoritmo original de Kennedy y Eberhart [9], la implementación de PSO utilizando vecindades estáticas converge más lentamente pero es menos vulnerable a atracción de mínimos locales en la función de adaptación [10, 11].

1.3. OPTIMIZACIÓN POR ENJAMBRES DE PARTÍCULAS (PSO)

Dentro de las vecindades estáticas, se experimentó con diversas estructuras de comunicación tomando en cuenta resultados de la psicología social incluyendo círculos, estrellas, conexiones aleatorias e incluso algunas con estructura de mundo pequeño. El resultado más importante es que aunque existen diferencias en el rendimiento para cada estructura de la vecindad, el análisis no es concluyente sobre cuál es en general mejor que la otra [10, 11]. Sin embargo, resultados recientes en el estudio de bandadas de pájaros estorninos (*Sturnus vulgaris*) señalan que cada estornino responde a una vecindad fija de entre seis y ocho agentes dentro de la bandada y con ello se optimiza el intercambio entre la cohesión del grupo y el esfuerzo individual [15–17]. Por ello, en esta tesis utilizamos una vecindad estática en la que cada partícula tiene exactamente 7 vecinos generados de forma aleatoria al inicio del algoritmo.

1.3.3. Inercia

En el modelo clásico de PSO (1995) [9], se utiliza una constante positiva V_{max} para limitar de forma arbitraria la velocidad de las partículas y mejorar la resolución de la búsqueda. Este factor V_{max} sirve como una restricción para controlar la habilidad de exploración global de una partícula. En 1998 Shi y Eberhart [12] introdujeron el concepto de *inercia* para tener un mejor control de la exploración del espacio de búsqueda. La motivación era eliminar la necesidad de la restricción V_{max} . El resultado fue una modificación a la ecuación de velocidad 1.4 que se convierte en la siguiente:

$$v_i(t + 1) = \omega v_i(t) + c_1 \gamma_{1i}(p_i(t) - x_i(t)) + c_2 \gamma_{2i}(p_g(t) - x_i(t)) \quad (1.6)$$

donde a ω se le denominó *factor de inercia*. Nótese que si interpretamos $c_1 \gamma_{1i}(p_i(t) - x_i(t)) + c_2 \gamma_{2i}(p_g(t) - x_i(t))$ como una fuerza externa F_i actuando en una partícula, entonces el cambio en la velocidad de la partícula (es decir la aceleración) puede escribirse como $\Delta v_i = F_i - (1 - \omega)v_i$. Así, la constante $(1 - \omega)$ actúa como un coeficiente de fricción y ω puede ser interpretada como la fluidez del medio en el que se mueve la partícula. Empíricamente se ha mostrado que los mejores resultados se obtienen asignando a ω valores relativamente altos al principio (aproximadamente 0.9), lo cual corresponde a un sistema en el que las partículas se mueven en un medio de baja viscosidad y realizan una exploración extensa, y reduciendo progresivamente el valor de ω a un valor pequeño (aproximadamente 0.4), con lo que el sistema tiende a explorar el espacio en torno a mínimos locales [10, 13].

El factor de inercia tiene un papel fundamental al balancear la exploración y la explotación dentro del proceso de búsqueda, pues determina la contribución de la velocidad previa de la partícula en el siguiente paso de tiempo. Debido a ello, diversas estrategias se desarrollaron para modificar la inercia durante el proceso de búsqueda: Inercia constante, aleatoria, adaptativa, sigmoidea creciente, sigmoidea decreciente, linealmente decreciente, caótica, oscilatoria, etc [10, 13]. La lista de posibles estrategias es interminable. En esta tesis utilizamos inercia linealmente decreciente debido a investigaciones de Bansal [13] que muestran que esta estrategia produce buenos

resultados en diversos problemas. En este modelo, la inercia es una función del paso de tiempo definida de la siguiente forma:

$$\omega(t) = M - \frac{M - m}{T}t \quad (1.7)$$

donde M es la inercia inicial en la primera iteración, m es la inercia final en la última iteración, t es la iteración actual y T es el número total de iteraciones. Finalmente, la ecuación 1.6 se convierte en

$$v_i(t + 1) = \omega(t)v_i(t) + c_1\gamma_{1i}(p_i(t) - x_i(t)) + c_2\gamma_{2i}(p_g(t) - x_i(t)) \quad (1.8)$$

Utilizando 1.8 y una adecuada elección de M , m , c_1 y c_2 , PSO se vuelve mucho más estable y el uso de la restricción V_{max} no afecta a la dinámica del sistema [10].

Al igual que los algoritmos genéticos, la optimización por enjambres de partículas se ha convertido en una herramienta eficaz en problemas de búsqueda y optimización. Particularmente, en esta tesis mostramos el uso de ambas técnicas para realizar búsquedas de parámetros que participan en la dinámica compleja de una red de regulación bioquímica ligada a la motilidad del espermatozoide del erizo de mar *Strongylocentrotus purpuratus* y que intenta modelar la dinámica del calcio intracelular en el flagelo del espermatozoide. Este estudio no solamente ayuda a comprender la forma en que los espermatozoides se aproximan a los gametos femeninos, sino también presenta un espacio de búsqueda geoméricamente complicado donde podemos estudiar la efectividad y comportamiento de algoritmos genéticos y optimización por enjambres de partículas.

1.4. Fecundación y redes de regulación bioquímica



Figura 1.4: La fecundación es uno de los problemas fundamentales en el estudio del desarrollo de organismos multicelulares.

1.4. FECUNDACIÓN Y REDES DE REGULACIÓN BIOQUÍMICA

En la naturaleza, el desarrollo de organismos multicelulares se basa en la diferenciación celular y patrones espaciotemporales. Dado que estos procesos surgen a partir de interacciones complejas entre elementos genéticos y no genéticos, modelos matemáticos y computacionales se desarrollaron para poder estudiarlos. Los modelos de redes de regulación genética ([18]), basados en datos experimentales, fueron capaces de reproducir exitosamente patrones de activación en genes que caracterizan diferentes tipos de células en plantas y animales. Dichos patrones corresponden a los atractores de estas redes y se han interpretado como el destino de las células [19].

En el mismo contexto, uno de los problemas fundamentales del desarrollo de organismos multicelulares es la fecundación. Entender cómo los espermatozoides se aproximan a los gametos femeninos ha sido por un largo tiempo un tema central en biología. Recientemente, diversos trabajos arrojaron evidencia experimental que apunta a la existencia de una relación entre las oscilaciones de la concentración intracelular de iones de calcio ($[Ca^{2+}]_i$) en el espermatozoide del erizo de mar y la motilidad de este. Así, determinar la estructura y dinámica de los elementos que producen dichas oscilaciones es una tarea fundamental. Diversos trabajos fueron realizados para proponer modelos fenomenológicos basados en consideraciones mecánicas e hidrodinámicas para describir el movimiento del espermatozoide. Estos modelos pudieron describir con éxito cómo la forma y las características en el flagelo determinan la dirección de nado. Sin embargo se desconoce cómo la concentración de $[Ca^{2+}]_i$ y la subyacente cadena de reacciones que lo controla, moldea la curvatura del flagelo. Basándose en experimentos realizados en el espermatozoide del erizo de mar *S. purpuratus*, como punto de partida en [20] se aborda el problema con una perspectiva molecular para proponer un modelo de señalización utilizando una red de regulación bioquímica en la que los nodos son variables discretas correspondientes a los elementos involucrados en la cadena de reacciones químicas y la transmisión de señales se lleva a cabo en intervalos de tiempo de acuerdo a reglas lógicas.

1.4.1. La red del calcio

En su búsqueda del gameto femenino, los espermatozoides del erizo de mar *Strongylocentrotus purpuratus* y *Lytechinus pictus* responden a *speract*, un decapeptido presente en la capa exterior del huevo que se difunde en el mar. Cuando el *speract* se adhiere con su receptor, el cual se encuentra en la membrana del flagelo en el espermatozoide, se inicia una cascada de señales que producen una serie de oscilaciones en las concentraciones del ion calcio intracelular $[Ca^{2+}]_i$. Resultados experimentales en dos dimensiones señalan que los espermatozoides nadan describiendo círculos. Sin embargo, cuando el flagelo detecta *speract*, los cambios en las fluctuaciones de $[Ca^{2+}]_i$ producen una curvatura más pronunciada seguida por un nado en línea recta [21]. Cuando el *speract* es reconocido por el receptor presente en la membrana del flagelo, dicho receptor activa la enzima de membrana *guanilato ciclasa* (GC) que produce *guanósín monofosfato cíclico* (cGMP). El incremento de cGMP abre un canal de potasio dependiente de cGMP (KCNG), lo cual produce una hiperpolarización en el potencial de la membrana V. Esto dispara los siguientes procesos:

1. La activación de un intercambiador Na^+/Ca^{2+} (NCE) que disminuye los ni-

veles flagelares de $[Ca^{2+}]_i$.

2. La activación de un intercambiador Na^+/H^+ (NHE) que incrementa el pH intracelular (pHi).
3. La activación de un canal catiónico activado por voltaje y dependiente de nucleótidos cíclicos (HCN).
4. Remoción de la inactivación de los canales de calcio voltaje dependientes de alto y bajo umbral (HVA y LVA).

El incremento en pHi disminuye la actividad de GC y activa una *adenilato ciclase soluble* (sAC) con la subsecuente producción de *adenosín monofosfato cíclico* (cAMP). Este último estimula un canal de calcio cAMP-dependiente (cAMPCC) y el previamente activado canal HCN, el cual tiende a repolarizar el potencial de membrana V. La repolarización abre los mencionados canales HVA y LVA causando una depolarización y un incremento en el calcio intracelular $[Ca^{2+}]_i$. Finalmente, para reinicializar la cadena se requiere una nueva hiperpolarización. Esta puede lograrse a través de un canal de cloro $Cl^- Ca^{2+}$ -dependiente (CaCC) y un canal de potasio $K^+ Ca^{2+}$ -dependiente (CaKC) que se abren cuando el calcio intracelular $[Ca^{2+}]_i$ es alto. Mecanismos constantes de extrusión de Ca^{2+} presentes en el flagelo, como bombas de calcio (CaP) y NCE mantienen niveles basales de $[Ca^{2+}]_i$. El mecanismo anterior se repite cíclicamente para generar una serie de oscilaciones de Ca^{2+} que produce una secuencia repetitiva de giros del esperma [20].

Este conjunto de interacciones puede resumirse en la figura 1.5:

1.4. FECUNDACIÓN Y REDES DE REGULACIÓN BIOQUÍMICA

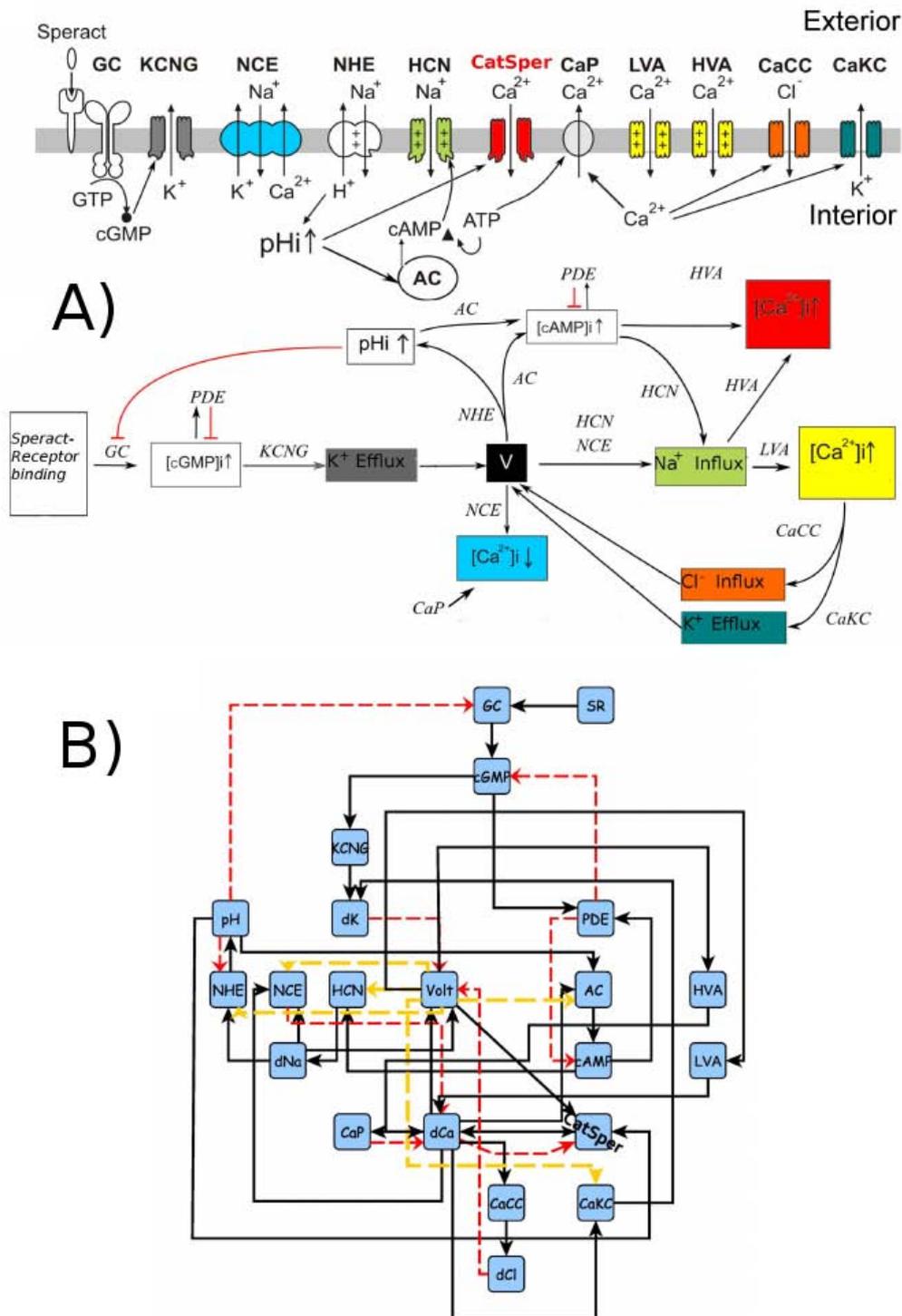


Figura 1.5: A) Vía de señalización para el calcio intracelular $[Ca^{2+}]_i$ en el flagelo del espermatozoide. B) Circuito lógico para la red de señalización en A.

1.4.2. Dinámica discreta

Para analizar la dinámica de la red descrita en la figura 1.5, se implementó una formulación discreta que generaliza una aproximación booleana, reveladora para la

dinámica de regulación en diversos sistemas [20].

En esta aproximación, el estado dinámico de la red consiste de un conjunto de N variables discretas x_1, x_2, \dots, x_N , donde cada una representa el estado de un nodo. Para la red descrita en la figura 1.5, la mayoría de las variables pueden tomar el valor 0 o 1, dependiendo de si el correspondiente elemento esta presente o ausente, abierto o cerrado, activo o inactivo, etc. Sin embargo, dos nodos requieren ser representados por variables de tres estados: el potencial de membrana V (hiperpolarizada 0, en descanso 1, depolarizada 2) y la concentración de calcio intracelular $[Ca^{2+}]_i$ (basal 0, tónico 1, supratónico 2). El estado de cada nodo x_n es determinado por su conjunto de reguladores (que son algunos otros nodos de la red). Se denota a $x_{n1}, x_{n2}, \dots, x_{nk}$ como los k reguladores de x_n . En cada paso de tiempo, el valor de x_n está dado por la siguiente ecuación [20, 22]:

$$x_n(t + \tau) = F_n(x_{n1}(t), x_{n2}(t), \dots, x_{nk}(t)) \quad (1.9)$$

donde F_n es una función de regulación construida tomando en cuenta la naturaleza activadora/inhibidora de los reguladores y cada nodo tiene su propia función de regulación [20].

Con esto se logró un mayor entendimiento del proceso de fecundación. Se hicieron predicciones que posteriormente fueron corroboradas experimentalmente y se avanzó en la comprensión de diversos problemas relevantes a la fertilización [20].

1.4.3. Dinámica de Glass

Para poder desarrollar un modelo continuo basado en ecuaciones diferenciales de la red del calcio, se requeriría conocer todas las constantes de reacción, afinidades promotoras, tazas de degradación y muchos otros parámetros involucrados en la dinámica. Hasta donde se sabe, algunos se conocen pero la mayoría no han sido identificados. Alternativamente, como paso hacia la descripción del modelo continuo de la red del calcio se puede implementar la dinámica de Glass en la red. Esto se logra considerando el parámetro τ en la ecuación 1.9 como una cantidad pequeña y desarrollando el lado izquierdo de esta ecuación en en serie de Taylor hasta el primer orden en τ [23]:

$$\frac{dx_n(t)}{dt} = \alpha_n [F_n(\hat{x}_{n1}(t), \hat{x}_{n2}(t), \dots, \hat{x}_{nk}(t)) - x_n(t)] \quad (1.10)$$

donde $\alpha_i = 1/\tau_i$ es el tiempo de relajación en el perfil de expresión de x_i . Los términos \hat{x}_{ni} hacen referencia al estado discreto del i -ésimo regulador del nodo x_n y se definen de la siguiente forma:

$$\hat{x}_n = H(x_n - \theta_n) \quad (1.11)$$

donde θ_n es un umbral y $H(x)$ es una función de paso que para variables dicotómicas se define como [23]:

1.4. FECUNDACIÓN Y REDES DE REGULACIÓN BIOQUÍMICA

$$H(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad (1.12)$$

De esta forma, la variable continua x_n representa el nivel de expresión del nodo n , mientras que \hat{x}_n representa el estado de expresión de dicho nodo: “Encendido” si x_n se encuentra sobre el umbral θ_n y “Apagado” si x_n está por debajo de θ_n . La ecuación 1.12 se generaliza para variables con más de dos estados [23].

Nos referimos a la dinámica generada por la ecuación 1.10 como *Dinámica de Glass* [23].

En esta tesis utilizamos algoritmos genéticos y optimización por enjambres de partículas para encontrar, por cada nodo i de la red, los valores de los parámetros α_n y θ_n con los que las ecuaciones de Glass reproducen de la mejor forma posible ciertas curvas experimentales de la concentración de calcio intracelular $[Ca^{2+}]_i$ como función del tiempo. Así, el espacio de búsqueda está formado por vectores de números reales de la forma $(\tau_1, \theta_1, \dots, \tau_n, \theta_n)$

CAPÍTULO 1. ANTECEDENTES

Capítulo 2

Algoritmos genéticos vs optimización por enjambres de partículas

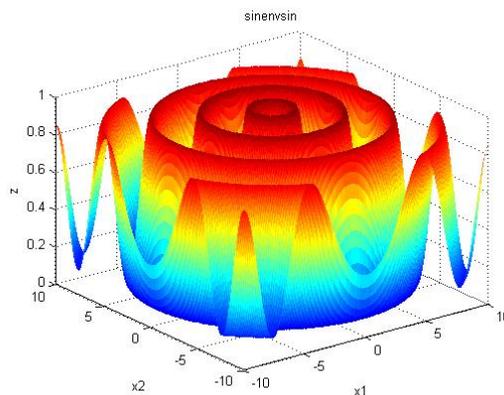


Figura 2.1: La función de Schaffer F6 es comunmente utilizada como función de prueba para evaluar diversas características de algoritmos de optimización debido a su gran cantidad de puntos máximos y mínimos

Dada la eficacia de los algoritmos genéticos y optimización por enjambres de partículas a través de los años, investigadores e ingenieros se han formulado diversas preguntas sobre cuál técnica debe ser utilizada para un problema específico. A pesar de los esfuerzos de diversos expertos en el área, ningún resultado parece concluir que en general una técnica es mejor que la otra o que cierto conjunto de funciones es optimizado de forma más eficiente por una de estas dos estrategias de búsqueda. Sin embargo, diversos investigadores se dieron a la tarea de encontrar diferencias y similitudes entre ambas en análisis teóricos y numéricos.

En este capítulo mostramos las conclusiones de algunos trabajos realizados por diversos investigadores comparando algoritmos genéticos y optimización por enjambres

de partículas desde diferentes puntos de vista y proporcionamos la metodología que utilizamos en esta tesis para comparar ambas técnicas.

2.1. Discusión previa

Algoritmos genéticos y optimización por enjambres de partículas se han comparado desde diferentes perspectivas. Estructuralmente, se realizaron investigaciones sobre el efecto de cada operador en ambas técnicas. Es decir, se estudiaron diferencias y similitudes entre la selección, cruce y mutación de los algoritmos genéticos y la dinámica de actualización de cada partícula considerando inercia y vecindades de partículas en PSO. Han sido analizados los tipos de problemas que cada técnica de optimización es capaz de resolver.

Por otra parte, la dinámica de ambas técnicas de optimización se ha comparado estudiando el balance entre intensificación y diversificación durante la ejecución de ambos algoritmos. La intensificación se refiere a la capacidad de explorar de manera local el espacio circundante a la mejor solución encontrada. La diversificación se refiere a la capacidad de explorar de manera global el espacio de búsqueda en su totalidad.

Finalmente, diversas comparaciones fueron realizadas desde el punto de vista numérico, observando los resultados obtenidos al utilizar ambas técnicas en la solución de un problema particular.

La revisión de los trabajos realizados para analizar las diferencias y similitudes entre algoritmos genéticos y optimización por enjambres de partículas nos permite proporcionar una metodología adecuada para comparar ambas técnicas en el caso de estudio propuesto en esta tesis.

Iniciando con la dinámica de ambas técnicas de optimización, Kachitvichyanukul [24] (2012) hace un estudio en el que se observa que en ambos algoritmos una de las principales diferencias está en el mecanismo para producir una nueva población de soluciones por medio de perturbaciones a los integrantes de la población anterior. Estos diferentes mecanismos generan una población de soluciones con distintos balances entre intensificación y diversificación. Para poder analizar ambas técnicas adecuadamente se asume que se utiliza la misma codificación para las soluciones y que la población inicial es exactamente la misma [24].

En este estudio se observa que en los algoritmos genéticos, las soluciones se evalúan en base a su función de adaptación. Los padres son seleccionados en base a probabilidades que favorecen individuos con un mejor valor de adaptación. La operación de cruce genera descendientes con elementos tomados de los padres y las soluciones tienden a parecerse a ellos. Basándose en esta observación, los algoritmos genéticos tienden a generar soluciones que probablemente convergen alrededor de distintas “buenas” soluciones en la población. La diversificación en los algoritmos genéticos se da a través del operador de mutación que produce perturbaciones en las solucio-

nes de vez en cuando [24].

Kachitvichyanukul señala que en contraste, en PSO el “nuevo” enjambre de partículas es generado por medio de las ecuaciones de velocidad 1.6 y actualización 1.5. Esto asegura que las nuevas partículas pueden ser muy diferentes a las anteriores. Además, la mejor posición visitada por el enjambre o la vecindad de una partícula influye en el comportamiento de todo el enjambre o toda la vecindad. Este comportamiento puede generar convergencia prematura alrededor de la mejor posición en la versión global de PSO [24].

Además, este estudio agrega que en los algoritmos genéticos la densidad de la población en el espacio de soluciones es menor, de tal forma que los operadores genéticos son incapaces de producir todas las soluciones potenciales, particularmente alrededor de la mejor solución. En optimización por enjambre de partículas la mejor solución tiene una fuerte influencia en el resto de las partículas y estas tienden a converger alrededor de ella explorando de forma más exhaustiva el espacio cercano [24].

Desde el punto de vista estructural, se realizaron investigaciones en ambos paradigmas sobre cómo cada operador afecta su comportamiento dentro del espacio de búsqueda. En ese sentido, una partícula en PSO es análoga a un cromosoma en un algoritmo genético y al igual que un cromosoma, una partícula representa una solución candidata al problema que se desea resolver [25].

Aunque existen diversas implementaciones de algoritmos genéticos, existe un consenso en que cualquier algoritmo genético tiene tres tipos de operadores: Selección, cruza y mutación. En PSO, no se etiqueta de esa forma a las operaciones que realiza, pero existen analogías. Eberhart y Shi (1998) [25] destacan el parecido entre ambas técnicas en términos de selección, cruza y mutación.

En un algoritmo genético, los cromosomas de la población son inicializados aleatoriamente, de tal forma que la cruza tiene efectos significativos al principio del algoritmo, moviendo a un cromosoma una distancia relativamente grande en el espacio de búsqueda. Hacia el final de la ejecución, la población converge de tal forma que muchos de los cromosomas tienen estructuras similares. En esta etapa, la cruza usualmente tiene un efecto menor y la distancia en el movimiento de los cromosomas es relativamente pequeña [25, 26].

En optimización por enjambres de partículas no se tiene la operación de cruza. Sin embargo, el concepto está presente debido a que cada partícula es acelerada con cierto grado de aleatoriedad hacia su mejor posición previa y hacia la mejor posición previa de toda la población o la vecindad de la partícula. Al igual que en los algoritmos genéticos, las partículas son inicializadas en posiciones aleatorias en el espacio de búsqueda y se desplazan distancias relativamente grandes debido al factor de inercia inicial ω , explorando el espacio de búsqueda. Hacia el final del algoritmo, ω toma valores más pequeños y las posiciones $pbest$ y $gbest$ influyen en la dirección de las partículas de forma más significativa, haciendo que converjan alrededor de

CAPÍTULO 2. ALGORITMOS GENÉTICOS VS OPTIMIZACIÓN POR ENJAMBRES DE PARTÍCULAS

mínimos locales, lo cual significa que la distancia que se mueven las partículas en cada iteración es relativamente pequeña [25].

En el caso de algoritmos genéticos, la cruce ocurre generalmente entre padres seleccionados aleatoriamente. La evolución de un cromosoma individual involucra el intercambio de material genético con algunos otros individuos seleccionados aleatoriamente. En optimización por enjambres de partículas, una partícula no intercambia material con otras partículas, pero su trayectoria se ve influenciada por ellas [25].

Los efectos de la mutación durante la ejecución de un algoritmo genético tienden a ser los opuestos de la cruce, pues la mutación tiende a tener un impacto menor al principio de la ejecución y mayor hacia el final. Dado que la población es inicializada aleatoriamente, cambiar un bit (codificación binaria) o un número real (codificación real) en algunas posiciones no representa un cambio significativo al inicio del algoritmo. Sin embargo, las mutaciones pueden ser significativas hacia el final de la ejecución aportando diversidad genética a una población que converge en cierto punto del espacio. Además, el mayor efecto de la mutación es que permite al algoritmo genético explorar todo el espacio de búsqueda al librar a los descendientes de las limitaciones en las características de los padres. En PSO, las partículas tienden a ser aceleradas hacia la mejor posición en la historia de la partícula (*pbest*) y hacia la mejor posición conocida por el enjambre (*gbest*). Sin embargo, los factores aleatorios γ_{1i} y γ_{2i} en la ecuación de velocidad 1.4 “perturban” la dirección de la partícula permitiéndole alcanzar potencialmente cualquier punto del espacio. Al igual que en algoritmos genéticos, estas perturbaciones tienen más impacto al final del algoritmo cuando el enjambre converge alrededor de mínimos locales que al principio del algoritmo cuando el enjambre tiene poca cohesión [25, 26].

En términos de selección, diversas estrategias pueden ser implementadas, pero su objetivo dentro de los algoritmos genéticos es promover la supervivencia del cromosoma mejor adaptado. La selección no existe en PSO pues todas las partículas permanecen como miembros de la población durante la ejecución del algoritmo. Sin embargo, la intención de preservar al individuo mejor adaptado en los algoritmos genéticos es influenciar el cambio de la población hacia él. Esta idea permanece en PSO en el sentido de que la mejor posición visitada por la partícula y la mejor posición visitada por el enjambre influyen en la trayectoria de la partícula [25, 26].

Algoritmos genéticos y enjambres de partículas también pueden ser comparados tomando en cuenta el tipo de problemas que pueden resolver. En ese sentido, Kennedy y Eberhart (1995) mencionan que:

Optimización por enjambres de partículas puede ser utilizada para resolver muchos problemas del mismo tipo de los que pueden ser resueltos por algoritmos genéticos [9]

Sin embargo, Eberhart y Kennedy (1995) estipulan que la optimización por enjambre de partículas no sufre de algunas dificultades presentadas por los algoritmos genéticos: La interacción en grupo permite a los enjambres de partículas mejorar el

2.2. METODOLOGÍA DE COMPARACIÓN

progreso hacia la solución óptima. Además, un sistema de enjambre de partículas tiene memoria de la cual carece un algoritmo genético. El cambio en la población de cromosomas resulta en una destrucción sobre el conocimiento previo del problema excepto cuando se utiliza elitismo, en cuyo caso solamente un pequeño número de individuos retienen sus identidades. En optimización por enjambres de partículas, los individuos que “vuelan” sobre un punto óptimo tienden a regresar hacia él y el conocimiento de las buenas soluciones se retiene por todas las partículas [11]. En la versión local de PSO, este conocimiento es retenido por lo menos por todas las partículas de la vecindad.

Finalmente, hemos mencionado que ambas técnicas de optimización pueden ser evaluadas desde el punto de vista numérico observando los resultados obtenidos por cada una al ser utilizadas en un problema particular. Li, Liu y Duan (2010) [26] encontraron que PSO presenta mejores resultados frente a un algoritmo genético al optimizar (en el sentido de la maximización) la función de Schaffer F6:

$$f(x, y) = 0,5 + \frac{\sin^2(\sqrt{x^2 + y^2}) - 0,5}{[1 + 0,001(x^2 + y^2)]^2} \quad (2.1)$$

Esta función es interesante al tener una infinidad de máximos y mínimos locales (figura 2.1). En este estudio, PSO tiene una tasa de convergencia más rápida hacia el óptimo global que los algoritmos genéticos. A su vez, PSO requiere menos tiempo de cómputo al contemplar operaciones más sencillas que los AG's. Además, en los resultados de este estudio se menciona que PSO requiere menos parámetros por parte del usuario y por ende es una técnica más sencilla de utilizar que los algoritmos genéticos.

Clerc (2004) [27] hace un estudio en el cual se aborda con buenos resultados el problema del agente viajero (TSP) mediante optimización por enjambres de partículas. Dada la naturaleza del problema, este estudio plantea modificaciones a las ecuaciones de velocidad 1.6 y actualización 1.5 para que el enjambre pueda moverse en un espacio de posiciones discreto y con restricciones. Esta particularidad para el problema TSP se convierte en una generalidad para diversos problemas donde el espacio de búsqueda tiene características similares. Es decir, existen problemas donde el algoritmo PSO debe sufrir modificaciones que incrementan la complejidad y disminuyen el rendimiento del algoritmo. En ese sentido, el algoritmo genético simple considera un espacio discreto de manera natural y requiere menos intervención por parte del usuario.

Chien-Ching Chiu, Yu-Ting Cheng y Chai-Wei Chang (2012) encontraron que la versión local de PSO obtiene mejores resultados que algoritmos genéticos al encontrar la mejor distribución de un conjunto de antenas de radio en un área con construcciones que interfieren y disminuyen el alcance de las antenas [28].

2.2. Metodología de comparación

CAPÍTULO 2. ALGORITMOS GENÉTICOS VS OPTIMIZACIÓN POR ENJAMBRES DE PARTÍCULAS

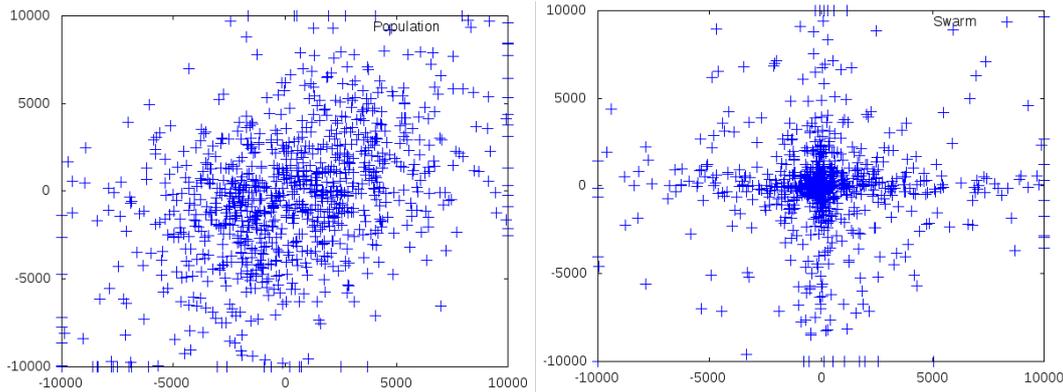


Figura 2.2: Izquierda: ubicación espacial de los cromosomas de un algoritmo genético en un estado avanzado de la ejecución para un problema dado. Derecha: ubicación espacial de las partículas de PSO en el mismo estado de ejecución que el algoritmo genético a la izquierda.

El objetivo de esta tesis es implementar una comparación numérica entre algoritmos genéticos y optimización por enjambres de partículas en un espacio de búsqueda con variables continuas. Para llevar a cabo dicho objetivo debe considerarse que ambas técnicas tienen que ser puestas a prueba bajo condiciones similares de ejecución. Basándonos en la descripción de algoritmos genéticos (sección 1.2) y optimización por enjambres de partículas (sección 1.3), así como de los aportes realizados con objetivos similares a los de este trabajo (capítulo 2.1), podemos proponer algunas características y restricciones que deben considerarse para comparar ambas técnicas. Adicionalmente, como caso de estudio para la aplicación de este objetivo elegimos abordar un problema relacionado con la fecundación externa en el erizo de mar *S. purpuratus*, lo cual permitirá avanzar en la comprensión de la forma en que el espermatozoide se aproxima al gameto femenino.

Consideramos que una de las características más importantes a tener en cuenta tiene que ver con la codificación del espacio de búsqueda. De manera natural, las partículas en PSO viajan en un espacio continuo por lo que la codificación de cada solución está representada por un vector de números reales [9]. Por otro lado, la versión simple de algoritmos genéticos considera una solución candidata como una cadena binaria, lo cual induce problemas en la precisión del resultado final, así como distancias de Hamming grandes entre soluciones cercanas [1, 3]. Debido a ello, consideramos que para una correcta y objetiva comparación entre algoritmos genéticos y enjambres de partículas, debe ser implementada la codificación real en algoritmos genéticos (sección 1.2.3). Al hacerlo, ambas técnicas realizan una búsqueda con una población que tiene la misma oportunidad potencial de viajar a un determinado punto del espacio. Es decir, los genes no están limitados a explorar solamente el espacio discreto inducido por la codificación binaria.

Para igualar condiciones en ambas técnicas, utilizamos elitismo en el operador de selección en algoritmos genéticos (el 10% de los individuos con el mejor valor de adaptación pasa a la siguiente generación) y utilizamos vecindades en el algoritmo

2.2. METODOLOGÍA DE COMPARACIÓN

PSO. Esto permite a ambos algoritmos considerar la historia de las mejores soluciones a lo largo de la ejecución y en ambos casos las mejores soluciones influyen en la evolución de la población.

Por otra parte elegimos el algoritmo de selección por torneo en la implementación de algoritmos genéticos pues tal como PSO, permite al algoritmo explotar el espacio cercano a las mejores soluciones y a la vez explorar el espacio lejano a estas [1].

Hemos decidido utilizar inercia decreciente en la implementación de enjambres de partículas. De esta forma, ambos algoritmos exploran el espacio de búsqueda total en las primeras iteraciones y convergen para explorar el espacio alrededor de un mínimo local hacia el final del algoritmo [12].

De igual forma, utilizamos mutación no uniforme en el algoritmo genético para que ambas técnicas tengan la oportunidad de generar elementos de la población lejanos a los mínimos locales encontrados en las últimas etapas de la ejecución [25,26].

Dado que hemos hecho un análisis comparativo entre el comportamiento de las partículas en PSO y los cromosomas en algoritmos genéticos, y dado que hemos elegido los operadores que creemos que acercan más a ambos algoritmos, consideramos que la comparación entre ellos debe llevarse a cabo con una población del mismo tamaño. Esto es, para poder evaluar objetivamente el desempeño de ambas técnicas, debe haber el mismo número de cromosomas en el algoritmo genético que partículas en PSO. De igual forma, debe considerarse el mismo número de iteraciones y la misma condición inicial (mismas posiciones de cromosomas y partículas) al llevar a cabo la comparación [24].

Al considerar la red de reacciones bioquímicas planteada en la sección 1.4, debemos notar que la dinámica descrita por las ecuaciones 1.10 de dicha red es extremadamente sensible a variaciones mínimas en los parámetros α_n y τ_n . Por ello, decidimos utilizar como caso de estudio la búsqueda de dichos parámetros, pues en este problema la forma geométrica de la función de adaptación muestra ser particularmente complicada, discontinua, con regiones planas, diversos máximos y mínimos locales y muy sensible a cambios en la posición de los agentes dentro del espacio de búsqueda. En la práctica, tanto algoritmos genéticos como enjambres de partículas fueron utilizados para resolver diversos problemas de optimización en espacios donde la función de adaptación es geoméricamente complicada. Sin embargo, consideramos que un problema con estas características es el mejor escenario para comparar la efectividad de ambos algoritmos.

Además de especificar el número de iteraciones y el tamaño de la población, tanto algoritmos genéticos como optimización por enjambres de partículas requieren la especificación de parámetros que intervienen en la ejecución. En algoritmos genéticos, debe especificarse la tasa de victoria del individuo con el mejor valor de adaptación en el operador de selección, la máxima perturbación que puede sufrir un individuo bajo mutación y un coeficiente de perturbación que define la forma en la distribución

CAPÍTULO 2. ALGORITMOS GENÉTICOS VS OPTIMIZACIÓN POR ENJAMBRES DE PARTÍCULAS

de probabilidad que produce mutaciones. Por otro lado, en el algoritmo PSO debe definirse la inercia inicial y final de las partículas así como el número de vecinos que cada partícula conoce. Debido a ello, diversos parámetros en uno y otro algoritmos deben ser evaluados y la comparación debe hacerse entre el mejor resultado arrojado por ambos.

Con base en lo visto en la sección 2.1 y dadas las consideraciones descritas anteriormente para algoritmos genéticos y optimización por enjambres de partículas, analizamos las diferencias entre ellos en los siguientes aspectos:

- **Evolución del valor de adaptación promedio en diversas ejecuciones.** Una forma de estudiar el comportamiento de la población tanto en algoritmos genéticos como en PSO es mediante el valor de adaptación promedio de la población en cada iteración. Dado que ambas técnicas de optimización se desarrollan de forma estocástica, es necesario analizar el valor de adaptación promedio tomando en cuenta más de una ejecución. Así, dado el valor de adaptación promedio $M_i(t)$ en la población durante la generación t en la i -ésima ejecución, el primer aspecto que nos interesa estudiar es la evolución del valor $S(t) = 1/L \sum_{i=1}^L M_i(t)$ en cada generación t , donde L es el número total de ejecuciones. Esto es el valor medio en cada generación de diversas ejecuciones del valor de adaptación promedio en la población. Este aspecto nos permite analizar la forma general con que la población converge hacia un mínimo local, así como la efectividad con la que cada técnica explora el espacio de búsqueda.
- **Evolución del valor de adaptación mínimo en diversas ejecuciones.** Al igual que el aspecto anterior, el seguimiento de la mejor solución encontrada a través de las iteraciones nos permite estudiar la efectividad con la que cada técnica explora el espacio de búsqueda y la rapidez con la que se acerca a la mejor solución encontrada. Como antes, debemos considerar la evolución de la mejor solución encontrada durante cada generación en diversas ejecuciones. Así, proponemos estudiar la media del valor de adaptación mínimo a través de las generaciones, obtenida a partir de un conjunto de ejecuciones con diferentes parámetros.
- **Evolución del valor de adaptación promedio en la mejor ejecución de cada técnica.** Esto es, la evolución del valor de adaptación promedio de la ejecución que arrojó los mejores resultados en ambas técnicas, a lo largo de las generaciones.
- **Evolución del valor de adaptación mínimo en la mejor ejecución de cada técnica.** Es decir, La evolución del valor de adaptación mínimo de la ejecución que arrojó los mejores resultados en ambas técnicas, a lo largo de las generaciones.
- **Mejor solución encontrada en diversas ejecuciones.** Dado que ambos son algoritmos estocásticos, diferentes ejecuciones pueden arrojar diferentes resultados en ambos algoritmos. Por ello, consideramos importante analizar el mejor resultado obtenido por cada uno dadas diferentes ejecuciones.

2.2. METODOLOGÍA DE COMPARACIÓN

Es importante mencionar que esta comparación no pretende calificar a un algoritmo como “mejor” sobre el otro, sino dar elementos de decisión que proporcionan un panorama sobre el comportamiento de ambas técnicas con el propósito de ayudar a elegir cuál de ellas utilizar.

La metodología propuesta en esta sección es utilizada en el siguiente capítulo en un caso de estudio que proporciona un espacio de búsqueda geoméricamente complicado y que pone a prueba las capacidades de algoritmos genéticos y optimización por enjambres de partículas para encontrar un conjunto de valores.

*CAPÍTULO 2. ALGORITMOS GENÉTICOS VS OPTIMIZACIÓN POR
ENJAMBRES DE PARTÍCULAS*

Capítulo 3

Caso de estudio “La red del calcio”



Figura 3.1: Ejemplar de erizo de mar *Strongylocentrotus purpuratus*

3.1. Contexto

Recordando la sección 1.4, uno de los problemas principales del desarrollo de los organismos multicelulares es entender cómo los espermatozoides se aproximan a los gametos femeninos. En ese sentido, se tiene evidencia experimental de que existe una relación entre las oscilaciones en la concentración intracelular de iones de calcio ($[Ca^{2+}]_i$) en el espermatozoide y la motilidad de este. En trabajos recientes, se propuso un conjunto de reacciones bioquímicas que inician con la adhesión de *speract* a su receptor y culminan con la oscilación de iones de calcio en el flagelo del espermatozoide. En un análisis posterior [29] se muestra que el modelo es más realista si se omiten los canales *HVA* y *LVA* y se incluye un canal de Ca^{2+} *pH* y voltaje dependiente llamado *CatSper*. La figura 3.2 muestra el circuito resultante de reacciones bioquímicas que se llevan a cabo en el flagelo del espermatozoide del erizo de mar, el cual se implementa en este modelo.

3.1. CONTEXTO

Uno de los estudios experimentales sobre el comportamiento de $[Ca^{2+}]_i$ en el flagelo de espermatozoide en el erizo de mar dio como resultado datos sobre la concentración de $[Ca^{2+}]_i$ a través de un corto periodo de tiempo una vez que el *speract* se adhiere a su receptor.

La figura 3.3 muestra una medición experimental de concentración de $[Ca^{2+}]_i$ en el flagelo del espermatozoide a lo largo del tiempo.

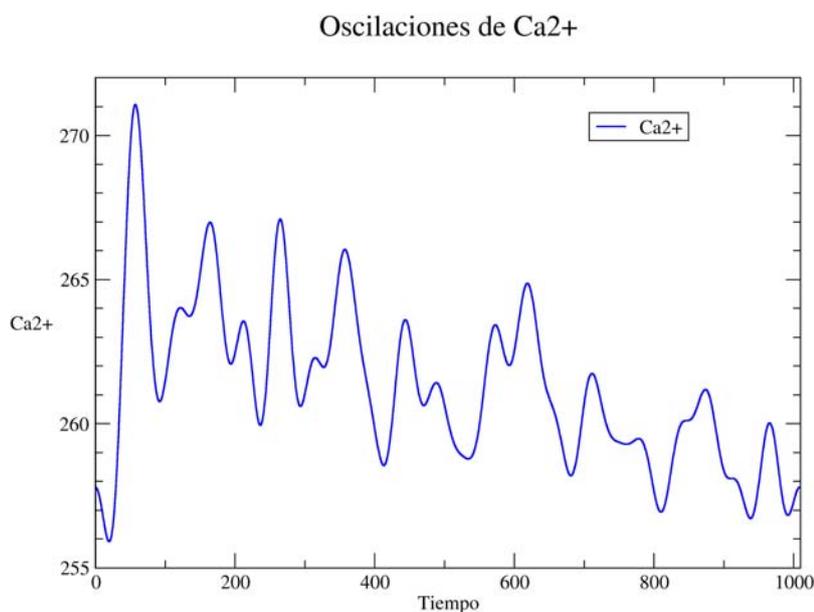


Figura 3.3: Oscilaciones de $[Ca^{2+}]_i$ a partir de la presencia de *speract*

Por otro lado, recordando la sección 1.4.3, la dinámica de Glass propuesta para modelar el estado de los nodos en la red del calcio nos dice que el estado de un nodo x_n en el tiempo t está dado por la ecuación 1.10:

$$\frac{dx_n(t)}{dt} = \alpha_n [F_n(\hat{x}_{n1}(t), \hat{x}_{n2}(t), \dots, \hat{x}_{nk}(t)) - x_n(t)] \quad (3.1)$$

donde $\alpha_i = 1/\tau_i$ es el tiempo de relajación en el perfil de expresión de x_i y

$$\hat{x}_n = H(x_n - \theta_n) \quad (3.2)$$

con θ_n un umbral y $H(x)$ como sigue:

$$H(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad (3.3)$$

Este planteamiento requiere la especificación de los valores $\alpha_i = 1/\tau_i$ y θ_i en cada nodo x_i para que la descripción de la red esté completa. Dadas las mediciones mostradas en la figura 3.3, se requiere encontrar aquellos valores α_i y θ_i para $i \in \{1, \dots, n\}$ que mejor reproducen el comportamiento de la figura 3.3 en el nodo Ca^{2+} del modelo de Glass. Esto es, deseamos aproximar las mediciones experimentales con el modelo de Glass y debemos encontrar los valores $\alpha_i = 1/\tau_i$ y θ_i para cada nodo de la red, que producen la mejor aproximación.

Hasta la escritura de esta tesis, no se cuenta con las mediciones experimentales de otros nodos de la red. Debido a ello se utiliza el comportamiento del modelo discreto 1.9 para aproximarlos, ya que este produjo buenos resultados.

Encontrar los valores específicos de los parámetros descritos es una tarea compleja que hasta donde sabemos no tiene solución analítica. Debido a ello, utilizamos algoritmos genéticos y optimización por enjambre de partículas para realizar esta tarea. El resto de este capítulo describe el procedimiento de aproximación realizado, así como los resultados obtenidos al utilizar ambas técnicas de búsqueda y optimización.

3.2. Transformación de los datos

Para realizar el estudio, las mediciones originales de la figura 3.3 deben ser reescaladas a un espacio más manejable.

Por una parte, sabemos que en el modelo discreto el nodo Ca^{2+} puede tomar tres posibles valores: basal 0, tónico 1, supratónico 2. Por ello, el estado de este mismo nodo en el modelo de Glass puede tomar valores en el intervalo $[0, 2]$. De la misma forma, el potencial de membrana V puede tomar uno de los siguientes tres valores en el modelo discreto: hiperpolarizada 0, en reposo 1, depolarizada 2. Así, el estado del nodo V en el modelo de Glass también puede tomar valores en el rango $[0, 2]$. El resto de los nodos pueden tomar los valores encendido 1 y apagado 0 en el modelo discreto, por lo que su estado en el modelo de Glass puede tomar valores en el rango $[0, 1]$.

Decidimos escalar las mediciones experimentales de la figura 3.3 al intervalo $[0, 2]$ donde sabemos que viven los valores del nodo Ca^{2+} en el modelo de Glass.

La escala de tiempo en la figura 3.3 está expresada en milisegundos. Sin embargo, por comodidad de los cálculos computacionales, decidimos reescalar este tiempo al intervalo $[0, 1]$.

Los datos resultantes se muestran en la figura 3.4

3.3. GENERANDO LAS SEÑALES FALTANTES

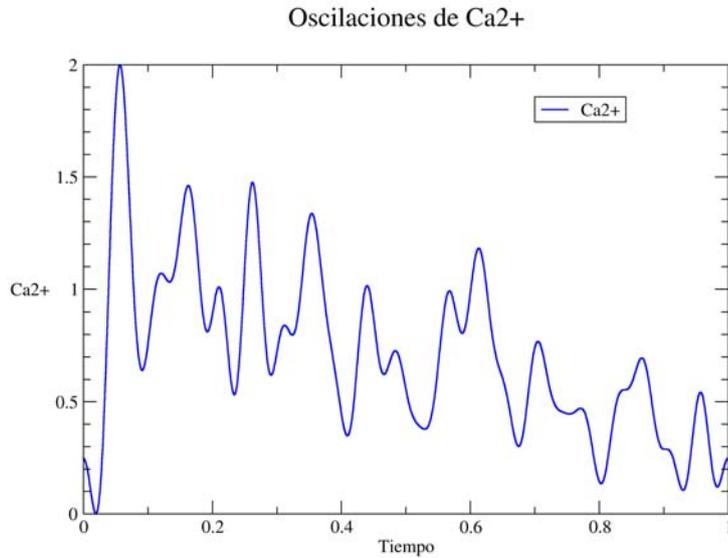


Figura 3.4: Mediciones experimentales de Ca^{2+} reescaladas

3.3. Generando las señales faltantes

Como hemos mencionado, carecemos de mediciones experimentales para el estado de los nodos de la red, excepto para Ca^{2+} . Sin embargo el modelo discreto mostró buenos resultados pues mediante este se realizaron predicciones que posteriormente fueron corroboradas experimentalmente. Dado que el modelo de glass está basado en la descripción del modelo discreto, consideramos como objetivo secundario de la aproximación mantener consistencia con el comportamiento de este último.

Dado que en el modelo discreto diferentes condiciones iniciales se desenvuelven en estados diferentes para cada nodo, utilizamos 10^6 condiciones iniciales diferentes para promediar las trayectorias generadas por cada una de ellas y así usar los promedios como una descripción más fina del estado de los nodos de la red a través del tiempo.

En la figura 3.5 cada curva es el promedio de 10^6 trayectorias generadas en cada nodo de la red a partir de condiciones iniciales distintas en el modelo booleano descrito en la sección 1.4.2 [20, 29].

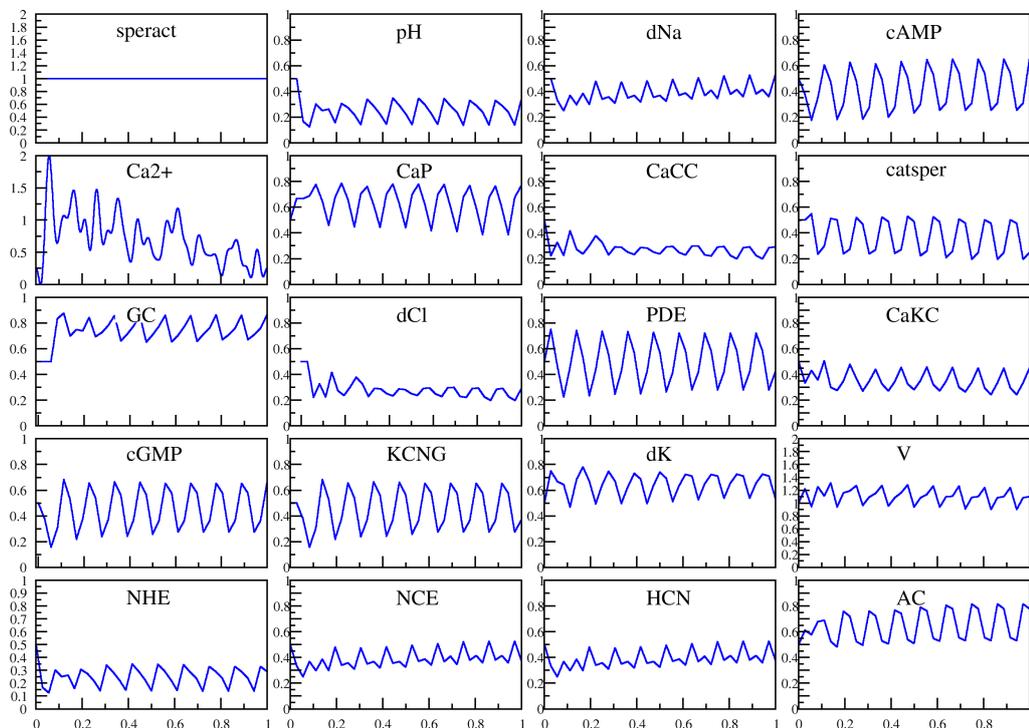


Figura 3.5: Cada curva es el promedio de 10^6 trayectorias generadas en cada nodo de la red a partir de condiciones iniciales distintas en el modelo booleano

3.4. Búsqueda de parámetros para la red del calcio

En este caso de estudio, utilizamos algoritmos genéticos de codificación real y optimización por enjambres de partículas para aproximar lo mejor posible el comportamiento de la dinámica de Glass (ecuación 1.10) a los resultados experimentales de la figura 3.4 y a las señales generadas de la figura 3.5. Para realizar dicha tarea, son necesarias las siguientes consideraciones:

- Denominamos **objetivos** a las señales mostradas en las figuras 3.4 y 3.5. El objetivo s_i corresponde a la señal especificada en el i -ésimo nodo de la red mostrada en la figura 3.2.
- Denominamos **aproximaciones** a las señales resultantes del modelo de Glass 3.1 utilizando un vector de parámetros $\bar{p} = (\tau_1, \theta_1, \dots, \tau_n, \theta_n)$.
- Denominamos $s_i(t)$ al valor del i -ésimo objetivo al tiempo t .

3.4. BÚSQUEDA DE PARÁMETROS PARA LA RED DEL CALCIO

- Denominamos $x_i^{\bar{p}}(t)$ al valor de la i -ésima aproximación (i -ésimo nodo de la red) al tiempo t con los valores τ_i y θ_i definidos en el vector \bar{p} .
- Denominamos *Cambio de pendiente* o *CDP* a un cambio en el signo de la derivada de una señal.

Las definiciones anteriores nos facilitan la explicación de la función de adaptación utilizada.

3.4.1. La función de adaptación

Dado que intentamos encontrar un conjunto de parámetros τ_i y θ_i para cada nodo x_i de la red, el espacio de búsqueda está formado por vectores de números reales de la forma $(\tau_1, \theta_1, \dots, \tau_n, \theta_n)$.

La función de adaptación utilizada tanto por algoritmos genéticos de codificación real como por optimización por enjambre de partículas considera los siguientes puntos:

- La diferencia entre el número de CDP's en las aproximaciones y el número de CDP's en los objetivos.
- La distancia de los puntos máximos y mínimos entre las aproximaciones y los objetivos

La idea de la función de adaptación es que los puntos máximos de las aproximaciones sean tan cercanos como sea posible a los puntos máximos de los objetivos. De manera análoga, buscamos la menor distancia entre los puntos mínimos de los objetivos y las aproximaciones. Por otro lado, creemos que el número de oscilaciones de las aproximaciones no requiere ser exactamente el número de oscilaciones de los objetivos para ser consideradas como "aproximaciones aceptables". Por ello, la función de adaptación está dividida en dos términos: diferencia en el número de CDP's y distancia entre ellas.

Dados el conjunto de objetivos $S = \{s_1, \dots, s_n\}$ y el conjunto de aproximaciones $X^{\bar{p}} = \{x_1^{\bar{p}}, \dots, x_n^{\bar{p}}\}$, denominamos $\Omega(\bar{p})$ como la diferencia entre el número de CDP's de S y el número de CDP's de $X^{\bar{p}}$ y la definimos como sigue:

$$\Omega(\bar{p}) = \sum_{i=0}^n \lambda_i \left| 1 - \frac{\omega(x_i^{\bar{p}})}{\omega(s_i)} \right| \quad (3.4)$$

donde $\omega(x)$ es el número de CDP's de la señal x y λ_i es un factor de ponderación para el nodo i de la red.

Por otra parte, sean $M_{s_{ij}}$ las coordenadas del j -ésimo máximo local del objetivo s_i , $M_{x_i^{\bar{p}}} = M_{x_i^{\bar{p}}1}, \dots, M_{x_i^{\bar{p}}k}$ el conjunto de coordenadas de los k máximos locales del nodo x_i con los parámetros τ_i y θ_i definidos en \bar{p} y sea $\delta(\bar{x}, \bar{y})$ la distancia euclidiana entre los vectores \bar{x} y \bar{y} . Definimos la función $\delta^M(M_{s_{ij}}, M_{x_i^{\bar{p}}})$ como sigue:

$$\delta^M(M_{s_{ij}}, M_{x_i^{\bar{p}}}) = \min \left\{ \delta(M_{s_{ij}}, M_{x_i^{\bar{p}l}}) \mid l \in \{1, \dots, k\} \right\} \quad (3.5)$$

Esta función representa la distancia entre el j -ésimo máximo local del objetivo s_i y el máximo local más cercano del nodo $x_i^{\bar{p}}$.

De forma análoga, definimos la función $\delta^m(m_{s_{ij}}, m_{x_i^{\bar{p}}})$ que representa la distancia entre el j -ésimo mínimo local del objetivo s_i y el mínimo local más cercano del nodo $x_i^{\bar{p}}$ como sigue:

$$\delta^m(m_{s_{ij}}, m_{x_i^{\bar{p}}}) = \min \left\{ \delta(m_{s_{ij}}, m_{x_i^{\bar{p}l}}) \mid l \in \{1, \dots, k\} \right\} \quad (3.6)$$

donde $m_{s_{ij}}$ es el j -ésimo mínimo local del objetivo s_i y $m_{x_i^{\bar{p}}} = m_{x_i^{\bar{p}1}}, \dots, m_{x_i^{\bar{p}k}}$ el conjunto de coordenadas de los k mínimos locales del nodo x_i con los parametros τ_i y θ_i definidos en \bar{p} .

Tomamos la suma de distancias δ^M y δ^m para cada máximo y mínimos locales del objetivo s_i como sigue:

$$Q(s_i, x_i^{\bar{p}}) = \sum_{j=1}^k \delta^M(M_{s_{ij}}, M_{x_i^{\bar{p}}}) + \delta^m(m_{s_{ij}}, m_{x_i^{\bar{p}}}) \quad (3.7)$$

Así, definimos la función $\Psi(\bar{p})$ que representa la distancia de los puntos máximos y mínimos entre los objetivos y las aproximaciones como sigue:

$$\Psi(\bar{p}) = \sum_{i=1}^n \lambda_i Q(s_i, x_i^{\bar{p}}) \quad (3.8)$$

Por tanto, la función de adaptación $F(\bar{p})$ del vector de parámetros \bar{p} se define de la siguiente forma:

$$F(\bar{p}) = \Psi(\bar{p}) + \Omega(\bar{p}) \quad (3.9)$$

Esta función toma siempre valores mayores o iguales a cero y el objetivo de algoritmos genéticos y optimización por enjambres de partículas es encontrar el vector de parámetros \bar{p} que minimizan el valor de F .

Durante la realización de este trabajo, se experimentó con diversas y muy variadas funciones de adaptación que tomaban en cuenta diversos aspectos como la distancia euclidiana entre las aproximaciones y los objetivos, la distancia euclidiana entre las derivadas, el número de oscilaciones, la distancia entre ellas y la diferencia en amplitud entre estas. Entre todas las funciones de adaptación probadas, la que generó mejores resultados es la presentada en la ecuación 3.4.2

3.4.2. Ejecución

Para efectos del objetivo de esta tesis, se realizaron diversas ejecuciones para cada técnica de optimización con las características mostradas en la tabla 3.4.2

3.4. BÚSQUEDA DE PARÁMETROS PARA LA RED DEL CALCIO

Parámetro	AG	PSO	Descripción
No. Ejecuciones	10	10	Ejecuciones en cada técnica
Tam. Población	1000	1000	Elementos (partículas/cromosomas) en la población
Generaciones	2000	2000	Generaciones en cada ejecución
Vecinos	NA	7	Cantidad de vecinos de cada partícula (PSO)
Inercia inicial	NA	0.7 - 0.9	Rango del valor de inercia en la primera generación en PSO
Inercia final	NA	0.01 - 0.1	Rango del valor de inercia en la última generación en PSO
PT	0.6 - 0.8	NA	En AG, es el rango de la probabilidad de que el cromosoma con el mejor valor de adaptación gane el torneo en la selección
PC	0.6 - 0.8	NA	En AG, es el rango de la probabilidad de dos cromosomas seleccionados lleven a cabo la operación cruza
η_m	1	NA	Factor de distribución de la mutación en los cromosomas de un AG (ecuación 1.2)
Radio de mutación inicial	0.5 - 0.7	NA	Rango del radio de mutación máxima Δ_{max} en la primera generación del AG (ecuación 1.3)
Radio de mutación final	0.05 - 0.1	NA	Rango del radio de mutación máxima Δ_{max} en la última generación del AG (ecuación 1.3)

Nos interesa que durante la búsqueda se de preferencia al comportamiento del calcio intracelular $[Ca^{2+}]_i$, seguido por el nodo V y por último al resto de los nodos. Esto es, la función de adaptación debe ser más susceptible a variaciones en las señales generadas por el calcio intracelular y el voltaje que a cualquier otro nodo de la red. Por ello, el parámetro de ponderación λ_i en las ecuaciones 3.4 y 3.8 toma el valor 0.5 para el nodo de calcio intracelular, 0.3 para el nodo de voltaje V y $\frac{0.2}{18}$ para el resto de los nodos en la red.

Los resultados obtenidos con estos parámetros de ejecución se muestran en la siguiente sección.

3.4.3. Resultados

La figura 3.6 presenta la media del valor de adaptación promedio de la población para PSO y algoritmos genéticos de codificación real. En ella se aprecia que desde etapas muy tempranas de las ejecuciones, los enjambres de partículas convergen más rápido hacia un mínimo local. En la etapa final de todas las ejecuciones se nota un valor de adaptación promedio inferior en PSO que en algoritmos genéticos.

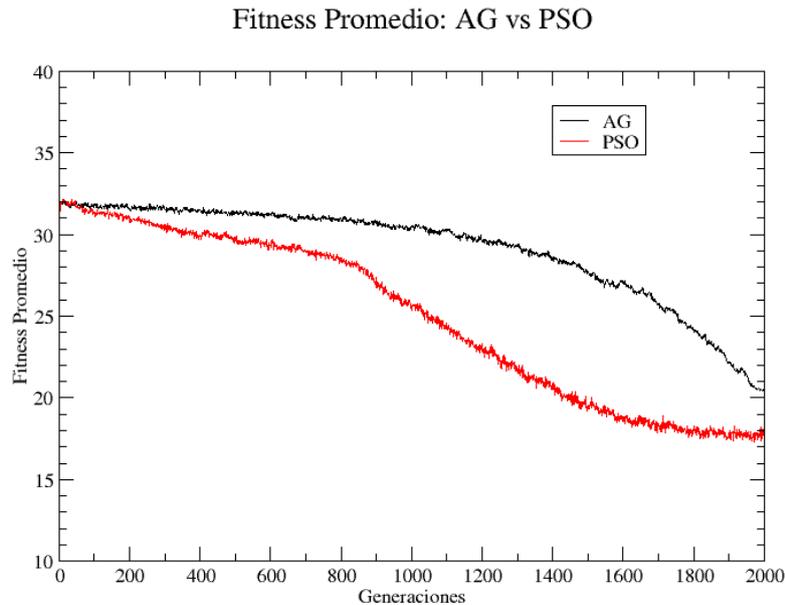


Figura 3.6: Media del valor de adaptación promedio para algoritmos genéticos y enjambres de partículas en 10 ejecuciones

Por otra parte, la figura 3.7 muestra la evolución del valor de adaptación mínimo promedio en ambas técnicas de optimización. En esta se ve que en general, PSO encuentra mejores soluciones que algoritmos genéticos desde etapas muy tempranas en las ejecuciones. Por otra parte, el comportamiento de ambas curvas muestra que tanto algoritmos genéticos como PSO convergen hacia un mínimo local en su mayor parte durante la primera mitad de las generaciones. Sin embargo, es notable que PSO sigue realizando exploraciones y ajustes finos durante la segunda mitad de las generaciones, mientras que los algoritmos genéticos convergen totalmente a un mínimo local durante la primera mitad de las generaciones. Las figuras 3.7 y 3.6 nos permiten deducir que el enjambre es más compacto que la población del algoritmo genético durante la segunda mitad de las generaciones. Debido a ello, los enjambres de partículas exploran de forma más eficaz el espacio alrededor de un mínimo local, lo cual les permite encontrar soluciones más acertadas.

3.4. BÚSQUEDA DE PARÁMETROS PARA LA RED DEL CALCIO

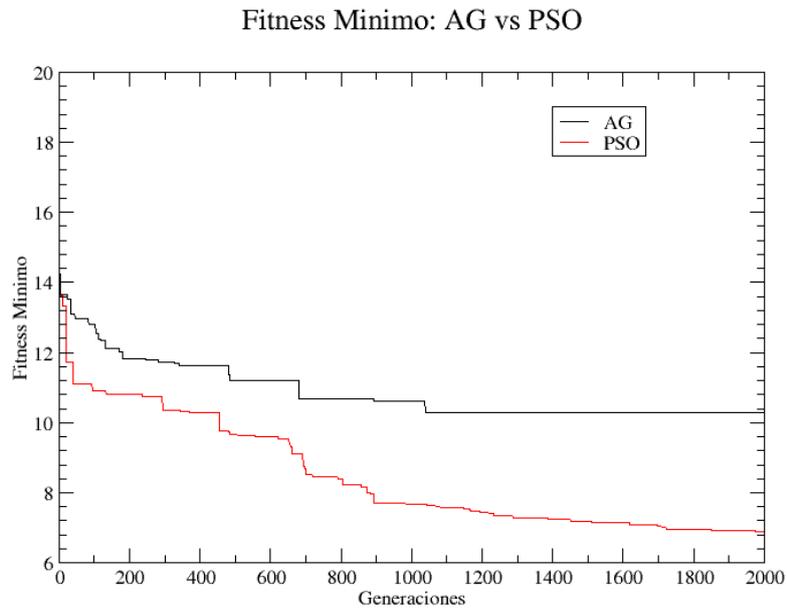


Figura 3.7: Media del valor de adaptación mínimo para algoritmos genéticos y enjambres de partículas en 10 ejecuciones

Dadas las consideraciones del capítulo 2.2, analizamos la ejecución que obtuvo los mejores resultados en ambas técnicas de optimización. La figura 3.8 muestra el valor de adaptación promedio de la población para algoritmos genéticos y optimización por enjambres de partículas. Aunque ambos terminan con un valor de adaptación promedio muy cercano, es notable que PSO mantiene desde etapas muy tempranas un valor de adaptación medio por debajo del algoritmo genético. De nuevo, esto muestra que el primero genera una población más compacta que el segundo.

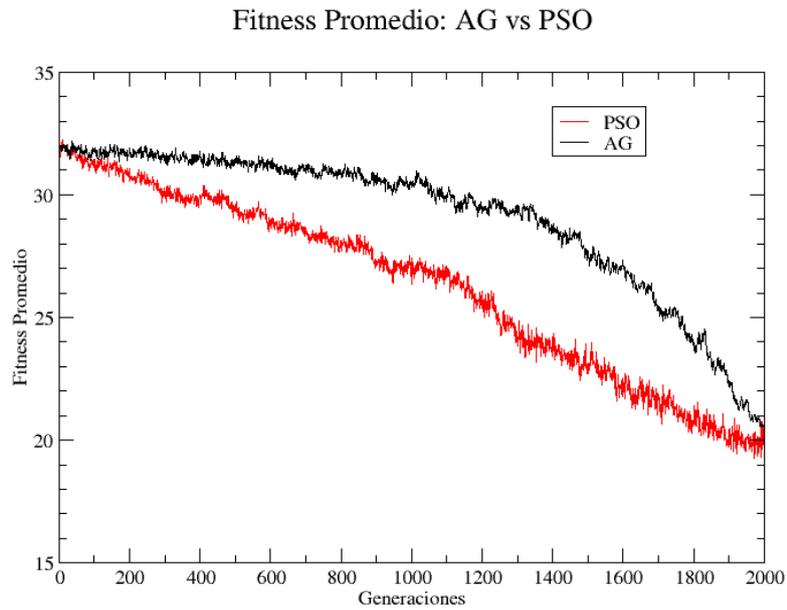


Figura 3.8: Valor de adaptación promedio para la mejor ejecución en ambas técnicas

La figura 3.9 muestra el comportamiento del valor de adaptación mínimo a lo largo de las generaciones para la mejor ejecución de algoritmos genéticos y optimización por enjambres de partículas. En esta gráfica se muestra que ambas técnicas describen un valor de adaptación mínimo muy cercano hasta la primera mitad de la ejecución de ambos algoritmos, con PSO ligeramente por debajo del algoritmo genético. Sin embargo al igual que antes, este último converge a su mejor solución antes de la mitad de las generaciones. Por el contrario, el enjambre de partículas continua la optimización durante la segunda mitad de la ejecución, obteniendo una mejor solución.

3.4. BÚSQUEDA DE PARÁMETROS PARA LA RED DEL CALCIO

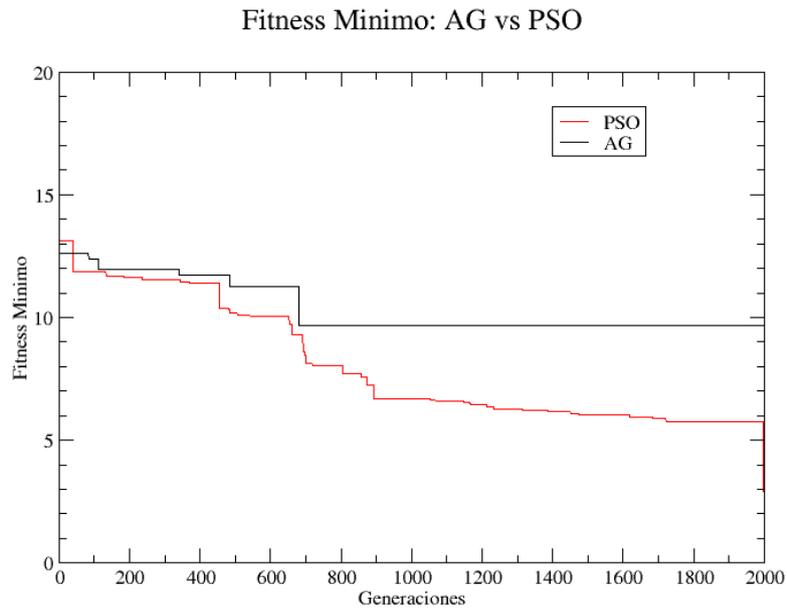


Figura 3.9: Valor de adaptación mínimo para la mejor ejecución en ambas técnicas

Finalmente, podemos comparar las mejores aproximaciones encontradas por ambas técnicas. La figura 3.10 muestra las mejores aproximaciones de algoritmos genéticos y optimización por enjambres de partículas a la señal del nodo $[Ca^{2+}]_i$ (figura 3.4). La figura 3.11 muestra la mejor aproximación de ambas técnicas a las señales del resto de los nodos en la red (figura 3.5).

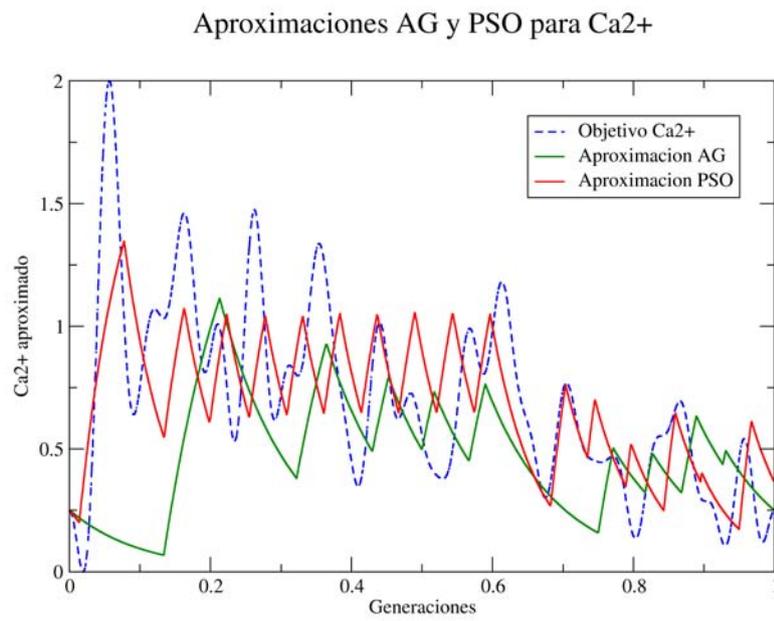


Figura 3.10: Mejor aproximación para el nodo Ca^{2+} en algoritmos genéticos y enjambres de partículas encontrados en un experimento

3.4. BÚSQUEDA DE PARÁMETROS PARA LA RED DEL CALCIO

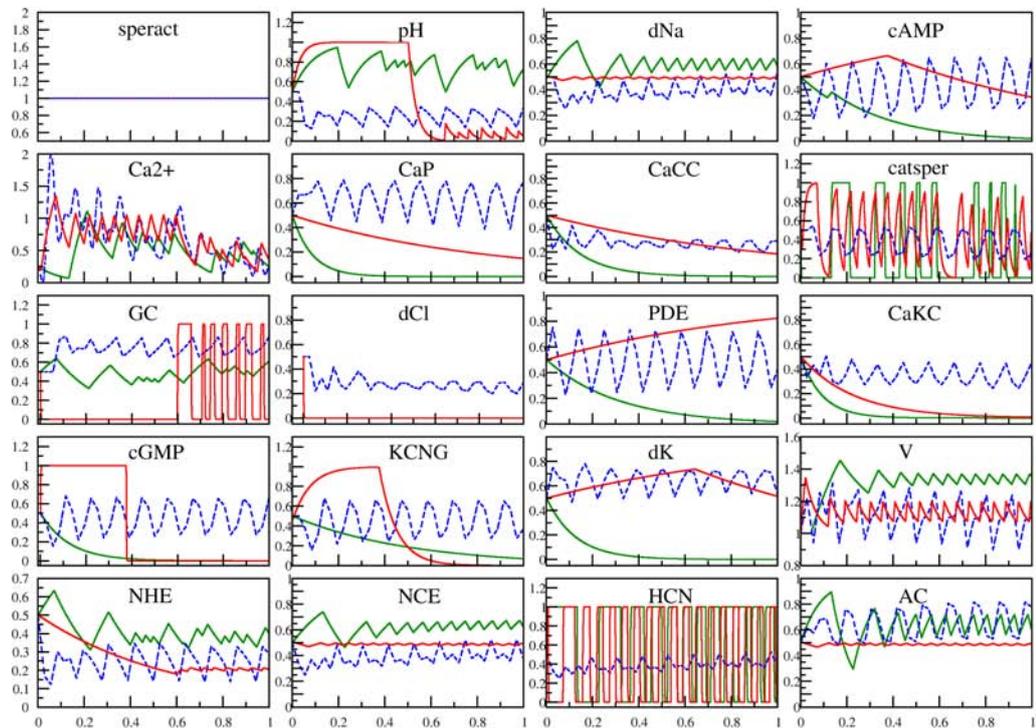


Figura 3.11: Mejor aproximación encontrada para todos los nodos de la red. Las curvas punteadas azules representan a los objetivos, las curvas verdes son las aproximaciones obtenidas por la mejor ejecución de los algoritmos genéticos de codificación real y las curvas rojas son las mejores aproximaciones obtenidas por la mejor ejecución de optimización por enjambres de partículas.

La tabla 3.4.3 muestra un resumen de los resultados obtenidos.

Atributo	AG	PSO
Media del valor de adaptación promedio en todas las ejecuciones	20.5095	17.8158
Media del valor de adaptación mínimo en todas las ejecuciones	10.2819	6.8889
Valor de adaptación promedio de la mejor ejecución	20.5560	19.9345
Valor de adaptación mínimo de la mejor ejecución	9.6670	5.7431

CAPÍTULO 3. CASO DE ESTUDIO “LA RED DEL CALCIO”

La evidencia mostrada señala una marcada diferencia entre el desempeño de algoritmos genéticos y optimización por enjambres de partículas, siendo este último el que obtuvo los mejores resultados para el problema de la red del calcio.

Capítulo 4

Conclusiones

En este trabajo de tesis hemos realizado una comparación numérica entre algoritmos genéticos y enjambres de partículas en un espacio de búsqueda de variables continuas. En el resto de este capítulo, enunciamos una serie de conclusiones que podemos obtener a partir de dicha comparación. Continuamos mencionando las contribuciones que este trabajo aporta a la ciencia. Finalmente, se menciona el trabajo futuro que puede surgir a partir de esta tesis.

Los resultados revisados en la sección 3.4.3 nos permiten realizar las siguientes conclusiones:

- **Optimización por enjambres de partículas tiene un mejor desempeño que los algoritmos genéticos de codificación real en el caso de estudio de la red del calcio.** Podemos concluir esto al analizar el comportamiento y resultado final de ambas técnicas, pues el valor de adaptación promedio de las partículas en PSO fue siempre inferior al de los cromosomas en AG's. Además, el mejor valor de adaptación siempre fue obtenido por PSO.
- **Optimización por enjambres de partículas explora el espacio de búsqueda de forma más eficiente que los algoritmos genéticos de codificación real.** La evidencia mostrada en la sección 3.4.3 nos muestra que PSO explora el espacio de búsqueda durante toda la ejecución del algoritmo; inicialmente en el espacio completo y con ajustes finos hacia el final de la ejecución. En cambio los algoritmos genéticos convergen a su mejor solución en la primera mitad de la ejecución, lo cual indica que PSO explora mejor el espacio de soluciones alrededor de un mínimo local.
- **La comparación entre algoritmos genéticos y optimización por enjambres de partículas es más clara cuando se realiza utilizando un problema real.** Existen reportes en la literatura de que los enjambres de partículas son capaces de resolver los mismos problemas que los algoritmos genéticos. Este juicio se realizó mostrando que ambas técnicas son capaces de encontrar de forma certera soluciones a funciones de adaptación que poseen un gran número de mínimos locales. Sin embargo la mayoría de estas funciones son continuas y diferenciables en todos sus puntos. Al revisar un caso de estudio como la red del calcio nos encontramos con un espacio de búsqueda

que tiene asociada una función de adaptación geoméricamente complicada y que es extremadamente sensible a perturbaciones en sus variables. Es en este tipo de problemas donde podemos llevar al límite las capacidades entre ambas técnicas y donde sus diferencias se hacen notables.

- A pesar de las conclusiones recién mencionadas, no pretendemos asegurar que optimización por enjambres de partículas es mejor que algoritmos genéticos en todos los problemas o en todos los espacios de búsqueda. Sin embargo podemos deducir que existe una alta probabilidad de que PSO obtenga los mejores resultados en un espacio de búsqueda cuya función de adaptación sea similar a la mostrada en el caso de estudio tratado en esta tesis: irregular con diversas discontinuidades, regiones planas y que es muy sensible a cambios en el dominio (puntos cercanos en el espacio de búsqueda pueden tener asociados valores de adaptación lejanos)

4.1. Contribuciones

Consideramos relevantes las siguientes contribuciones de este trabajo a las ciencias de la computación:

- El estudio elaborado en esta tesis provee elementos de decisión para elegir algoritmos genéticos de codificación real u optimización por enjambres de partículas ante determinados problemas. Particularmente, los resultados obtenidos sugieren utilizar este último en problemas de variables continuas, donde la función de adaptación presenta geometrías complicadas y puntos cercanos en el espacio de búsqueda pueden presentar valores de adaptación distantes.
- La metodología propuesta en el capítulo 2.2 mostró una forma de comparar numéricamente el desempeño de ambas técnicas de optimización no solamente en el resultado final o el mejor valor de adaptación, sino en la evolución completa de ambos algoritmos. El análisis de la ejecución completa nos permitió deducir la capacidad de ambas técnicas para explorar el espacio, así como la velocidad y eficacia para explorar mínimos locales. Esta metodología es aplicable a otras técnicas estocásticas de optimización basadas en agentes.

De manera colateral, los resultados mostrados en el caso de estudio tratado ayudan a decidir si el modelo de Glass es adecuado para describir la cadena de reacciones producida en el flagelo del espermatozoide en el erizo de mar *S. purpuratus*. El conocimiento aportado en esta área nos permite entender mejor la motilidad del esperma y en general, el problema de la fecundación.

4.2. Trabajo futuro

El trabajo realizado en esta tesis puede extenderse en diversas áreas de las cuales consideramos las siguientes:

4.2. TRABAJO FUTURO

- Optimización por enjambres de partículas se concibió como una técnica de optimización para espacios continuos. Por otra parte, los algoritmos genéticos simples se concibieron para explorar espacios discretos. En esta tesis, utilizamos una adaptación del algoritmo genético simple para explorar un espacio continuo. Consideramos que parte del trabajo futuro consiste en Adaptar PSO para explorar espacios discretos y realizar la misma comparación planteada en esta tesis utilizando casos de estudio como el problema de las N reinas o el problema TSP.
- En la implementación del algoritmo genético de codificación real descrito en esta tesis se eligieron los operadores de selección, cruza y mutación que consideramos son los más adecuados para optimizar una función de adaptación cuyas variables existen en un espacio continuo. Sin embargo consideramos importante extender la comparación realizada en esta tesis utilizando implementaciones de algoritmos genéticos que utilicen otros operadores genéticos. Particularmente, consideramos relevante la implementación de los operadores de cruza y mutación del algoritmo genético *BGA* [7] y una variante de este que utiliza *recombinación difusa* como algoritmo de cruza [30].
- Consideramos importante comparar PSO contra una técnica de optimización como *evolución diferencial* [31, 32], que muestra buenos resultados y fue diseñada desde su origen para resolver problemas de variables continuas. La comparación entre estas dos técnicas utilizando la metodología y caso de estudio propuestos en esta tesis nos permite obtener elementos de decisión sobre cuál técnica debe ser utilizada en problemas con funciones de adaptación similares a la evaluada en este trabajo.
- En el problema de la motilidad del esperma y la dinámica de Glass para la red del calcio, consideramos importante modificar el modelo para pasar a una dinámica continua en la cuál se puedan realizar búsquedas de nuevos parámetros. Este trabajo actualmente está en desarrollo.
- Aunque los mejores resultados numéricamente fueron obtenidos por enjambres de partículas, el comportamiento global del sistema se presenta mejor en algoritmos genéticos, por lo cual la función de adaptación propuesta en la sección 3.4.1 debe modificarse para obtener mejores resultados añadiendo valores de ponderación para cada nodo, que pueden ser ajustados conforme se obtengan mediciones experimentales para nodos distintos al Ca^{2+} .
- La señal del nodo Ca^{2+} corresponde a un solo experimento mientras que el resto de las señales generadas muestran el comportamiento promedio del modelo discreto. En un desarrollo futuro es posible considerar mediciones de diversos experimentos no solamente para el nodo Ca^{2+} sino para el resto de los nodos de la red.

Estas son solamente algunos de los aspectos que pueden ser desarrollados a partir de esta tesis.

CAPÍTULO 4. CONCLUSIONES

Bibliografía

- [1] Godfrey C. Onwubolu and B. V. Babu. *New optimization techniques in engineering*. Springer-Verlag Berlin Heidelberg, 2004.
- [2] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*, volume 412. Addison-wesley., 1989.
- [3] Melanie Mitchell. *An Introduction to Genetic Algorithms*. A Brandford book, 1996.
- [4] David E. Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.
- [5] Uday K. Chakraborty and Cezary Z. Janikow. An analysis of gray versus binary encoding in genetic search. *Inf. Sci.*, 156(3-4):253–269, November 2003.
- [6] J. David Schaffer, Richard A. Caruana, Larry J. Eshelman, and Rajarshi Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 51–60, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.
- [7] Dirk Schlierkamp-Voosen and Heinz Mühlenbein. Strategy adaptation by competing subpopulations. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature — PPSN III*, volume 866 of *Lecture Notes in Computer Science*, pages 199–208. Springer Berlin Heidelberg, 1994.
- [8] Ram Bhusan Agrawal, Kalyanmoy Deb, Kalyanmoy Deb, and Ram Bhushan Agrawal. Simulated binary crossover for continuous search space. Technical report, 1994.
- [9] J. Kennedy and R. Eberhart. Particle swarm optimization. *Proceedings of IEEE international conference on neural networks*, 4(2):1942–1948, 1995.
- [10] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. an overview. *Swarm Intelligence*, 1(1):33–57, 2007.
- [11] J. Kennedy and R. Eberhart. A new optimizer using particle swarm theory. proceedings of the sixth international symposium on. *Micro Machine and Human Science*, pages 39–43, 1995.

- [12] Y. Shi and R. Eberhart. A modified particle swarm optimizer. *Evolutionary Computation Proceedings*, pages 69–73, 1998.
- [13] J.C. Bansal, P.K. Singh, M. Saraswat, A. Verma, S.S. Jadon, and A. Abraham. Inertia weight strategies in particle swarm optimization. In *Third World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pages 633–640, Salamanca, Oct 2011. IEEE.
- [14] Chun-man Yan, Bao-long Guo, and Xian-xiang Wu. Empirical study of the inertia weight particle swarm optimization with constraint factor. *International Journal of Soft Computing And Software Engineering (JSCSE)*, 2(2), 2012.
- [15] George F. Young, Luca Scardovi, Andrea Cavagna, Irene Giardina, and Naomi E. Leonard. Starling flock networks manage uncertainty in consensus at low cost. *PLoS Comput Biol*, 9(1):e1002894, 01 2013.
- [16] Yilun Shang and Roland Bouffanais. Influence of the number of topologically interacting neighbors on swarm dynamics. *Scientific Reports*, 4, 02 2014.
- [17] W. Bialek, A. Cavagna, I. Giardina, T. Mora, E. Silvestri, M. Viale, and A. M. Walczak. Statistical mechanics for natural flocks of birds. *Proceedings of the National Academy of Science*, 109:4786–4791, mar 2012.
- [18] Sui Huang. The molecular and mathematical basis of waddington’s epigenetic landscape: A framework for post-darwinian biology? *Bioessays*, 34(2):149–157.
- [19] Sui Huang. Reprogramming cell fates: reconciling rarity with robustness. *Bioessays*, 31(5):546–560.
- [20] Espinal-Enríquez J, Darszon A, Guerrero A, and Martínez-Mekler G. In silico determination of the effect of multi-target drugs on calcium dynamics signaling network underlying sea urchin spermatozoa motility. *PLoS ONE*, 9(8), 2014.
- [21] CD Wood, T Nishigaki, Y Tatsu, SA Baba, M Whitaker, and A Darszon. Altering the speract-induced ion permeability changes that generate flagellar Ca^{2+} spikes regulates their kinetics and sea urchin sperm motility. *Dev Biol*, 306(2):525–37.
- [22] Espinal-Enríquez J, Darszon A, and Martínez-Mekler G. Robustness and redundancy in calcium signaling network relevant to marine sperm fertilization.
- [23] Álvarez Buylla ER, Chaos Á, Aldana M, Benítez M, Cortes-Poza Y, et al. Floral morphogenesis: Stochastic explorations of a gene network epigenetic landscape. *PLoS ONE*, 3(11), 2008.
- [24] Voratas Kachitvichyanukul. On comparison of three evolutionary algorithms: Ga, pso and de. *Industrial Engineering and Management Systems*, 11(3):215–223, 2012.

BIBLIOGRAFÍA

- [25] Russell C. Eberhart and Yuhui Shi. Comparison between genetic algorithms and particle swarm optimization. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, editors, *Evolutionary Programming VII*, volume 1447 of *Lecture Notes in Computer Science*, pages 611–616. Springer Berlin Heidelberg, 1998.
- [26] Zhijie Li, Xiangdong Liu, Xiaodong Duan, and Feixue Huang. Comparative research on particle swarm optimization and genetic algorithm. *Computer and Information Science*, 3(1):120–127, 2010.
- [27] M. Clerc. *New Optimization Techniques in Engineering*, chapter 8. Springer, 2004.
- [28] Chien-Ching Chiu, Yu-Ting Cheng, and Chai-Wei Chang. Comparison of particle swarm optimization and genetic algorithm for the path loss reduction in an urban area. *Journal of Applied Science and Engineering*, 15(4):371–380, 2012.
- [29] J. Espinal-Enriquez, A Darszon, C Beltrán, and G. Martinez-Mekler. Network model predicts that catsper is the main ca²⁺ channel in the regulation of sea urchin sperm motility. *por ser enviado a PLOS Computational Biology*.
- [30] H.-M. Voigt, H. Mühlenbein, and D. Cvetkovic. Fuzzy recombination for the breeder genetic algorithm. In *Proc. Sixth Int. Conf. on Genetic Algorithms*, pages 104–111. Morgan Kaufmann Publishers, 1995.
- [31] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, 1995.
- [32] Rainer Storn and Kenneth Price. Differential evolution; a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, dec 1997.