

## Una introducción a la computación evolutiva y algunas de sus aplicaciones en Economía y Finanzas

SANTANA QUINTERO, LUIS VICENTE

Departamento de Computación. CINVESTAV-IPN (Grupo de Computación Evolutiva)

Correo electrónico: [lvspenny@hotmail.com](mailto:lvspenny@hotmail.com)

COELLO COELLO, CARLOS A.

Departamento de Computación. CINVESTAV-IPN (Grupo de Computación Evolutiva)

Correo electrónico: [ccoello@cs.cinvestav.mx](mailto:ccoello@cs.cinvestav.mx)

### RESUMEN

Este artículo pretende proporcionar un panorama general de la computación evolutiva, sus orígenes, sus paradigmas principales y algunas de sus aplicaciones en Economía y Finanzas. Se discuten, entre otras cosas, los descubrimientos científicos más importantes que originaron el denominado Neo-Darwinismo, que es la teoría en la que se basa la computación evolutiva. También se proporciona una breve cronología de acontecimientos clave que desembocaron en los tres paradigmas en uso más común dentro de la computación evolutiva moderna: los Algoritmos Genéticos, la Programación Evolutiva y las Estrategias Evolutivas. En la segunda parte del artículo se proporcionan algunas aplicaciones representativas del uso de algoritmos evolutivos en Economía y Finanzas, así como algunas de las tendencias de investigación en el Área.

**Palabras clave:** algoritmos evolutivos; algoritmos genéticos; programación evolutiva; estrategias evolutivas.

**Clasificación JEL:** C61; C63.

**2000MSC:** 90C59; 91E28; 68T20.

# An Introduction to Evolutionary Computation and Some of its Applications in Economics and Finance

## ABSTRACT

This paper attempts to provide a general view of evolutionary computation, its origins, its main paradigms, and some of its applications in Economics and Finance. Among other topics, we discuss the most important scientific discoveries that originated the so called Neo-Darwinism, which is the theory on which evolutionary computation is based. We also provide a brief chronology of the key facts that culminated in the three paradigms in most common use within evolutionary computation today: genetic algorithms, evolutionary programming and evolution strategies. In the second part of the paper, we provide some applications that are representative of the use of evolutionary algorithms in Economics and Finance, as well as some of the research trends within this area.

**Keywords:** evolutionary algorithms; genetic algorithms; evolutionary programming; evolution strategies.

**JEL classification:** C61; C63.

**2000MSC:** 90C59; 91E28; 68T20.



# 1. Introducción

Desde los años 1930s, algunos investigadores comenzaron a ver el proceso de evolución de las especies como un proceso de aprendizaje, mediante el cual la naturaleza dota a las especies de diferentes mecanismos, buscando hacerlas más aptas para sobrevivir. Partiendo de estos preceptos no resulta entonces difícil percatarse de que pueden desarrollarse algoritmos que traten de resolver problemas de búsqueda y optimización guiados por el principio de la “supervivencia del más apto” que postulara Charles Darwin en su famosa (y controversial) teoría de la evolución de las especies. Dichos algoritmos son denominados, hoy en día, algoritmos evolutivos y su estudio conforma la computación evolutiva, que es el tema del que se ocupa este artículo.

Los algoritmos evolutivos, en sus diferentes variantes, han resultado muy exitosos en la solución de una amplia gama de problemas del mundo real. Nuestro objetivo principal es motivar el interés de los especialistas en economía y finanzas por utilizar algoritmos evolutivos. Para ello, además de describir brevemente su funcionamiento y sus ventajas principales, proporcionamos algunos ejemplos de su uso en economía y finanzas.

El resto del artículo está organizado de la siguiente forma. La sección 2 describe brevemente el Neo-Darwinismo, que es la teoría en la que se basa la computación evolutiva. En la sección 3 se proporciona una breve semblanza histórica de la computación evolutiva. Los 3 paradigmas principales que conforman la computación evolutiva se describen en la sección 4, incluyendo el algoritmo básico de cada uno y algunas de sus principales aplicaciones. En la sección 5 se describen algunas de las ventajas principales de los algoritmos evolutivos con respecto a otras técnicas de búsqueda y optimización. En la sección 6 se abordan algunos ejemplos representativos del uso de los algoritmos evolutivos en la economía y las finanzas. Finalmente, nuestras conclusiones se presentan en la sección 7.

## 2. Neo-Darwinismo

La teoría evolutiva propuesta originalmente por Charles Darwin en combinación con el seleccionismo de August Weismann y la genética de Gregor Mendel, se conoce hoy en día como el paradigma Neo-Darwiniano. El Neo-Darwinismo establece que la historia de la vasta mayoría de la vida en nuestro planeta puede ser explicada a través de una serie de procesos estadísticos que actúan sobre y dentro de las poblaciones y especies [40]: la reproducción, la mutación, la competencia y la selección.

La reproducción es una propiedad obvia de todas las formas de vida de nuestro planeta, pues de no contar con un mecanismo de este tipo, la vida misma no tendría forma de producirse. En cualquier sistema que se reproduce a sí mismo continuamente y que está en constante equilibrio, la mutación está garantizada [27]. El contar con una cantidad finita de espacio para albergar la vida en la Tierra garantiza la existencia de la competencia. La selección se vuelve la consecuencia natural del exceso de organismos que han llenado el espacio de recursos disponibles. La evolución es, por lo tanto, el resultado de estos procesos estocásticos (es decir,

probabilísticos) fundamentales que interactúan entre sí en las poblaciones, generación tras generación.

### 3. Una breve historia de la Computación Evolutiva

La evolución natural fue vista como un proceso de aprendizaje desde los 1930s. Walter D. Cannon, por ejemplo, plantea en su libro *The Wisdom of the Body* [19] que el proceso evolutivo es algo similar al aprendizaje por ensayo y error que suele manifestarse en los humanos. El célebre matemático inglés Alan Mathison Turing reconoció también una conexión “obvia” entre la evolución y el aprendizaje de máquina en su artículo (considerado hoy clásico en Inteligencia Artificial) titulado “*Computing Machinery and Intelligence*” [65].

A fines de los 1950s y principios de los 1960s, el biólogo Alexander S. Fraser [31, 32, 33] publicó una serie de trabajos sobre la evolución de sistemas biológicos en una computadora digital, dando la inspiración para lo que se convertiría más tarde en el algoritmo genético [41]. El trabajo de Fraser incluye, entre otras cosas, el uso de una representación binaria, de un operador de cruce probabilístico, de una población de padres que generaban una nueva población de hijos tras recombinarse y el empleo de un mecanismo de selección. Además, Fraser estudió el efecto de la epístasis<sup>1</sup>, la segregación<sup>2</sup>, los porcentajes de cruce y varios otros mecanismos biológicos que hoy son de sumo interés para la comunidad de computación evolutiva. Su trabajo de más de 10 años en este tema se resume en un libro titulado *Computer Models in Genetics* [34]. De tal forma, puede decirse que el trabajo de Fraser anticipó la propuesta del algoritmo genético simple de John Holland y la de la estrategia evolutiva de dos miembros de Hans-Paul Schwefel [26]. Fraser además llegó a utilizar el término “aprendizaje” para referirse al proceso evolutivo efectuado en sus simulaciones, y anticipó el operador de inversión, la definición de una función de aptitud y el análisis estadístico de la convergencia del proceso de selección [30].

A lo largo del siglo XX, diversos investigadores desarrollaron algoritmos inspirados en la evolución natural para resolver diferentes tipos de problemas. Por ejemplo, R. M. Friedberg [24,35,39] fue pionero en la tarea de evolucionar programas de computadora. El trabajo de Friedberg consistió en generar un conjunto de instrucciones en lenguaje máquina que pudiesen efectuar ciertos cálculos sencillos (por ejemplo, sumar dos números) [26]. Fogel [26] considera que Friedberg fue el primero en enunciar de manera informal los conceptos de *paralelismo implícito*<sup>3</sup> y *esquemas*<sup>4</sup>, que popularizara Holland en los 1970s [41].

---

<sup>1</sup> En biología se dice que un gen es “epistático” cuando su presencia suprime el efecto de un gen que se encuentra en otra posición.

<sup>2</sup> Cuando se forman los gametos y tenemos más de un par de cromosomas en el genotipo, entonces, para fines de la reproducción sexual, es necesario elegir uno de los cromosomas existentes. A este proceso se le llama *segregación*.

<sup>3</sup> El paralelismo implícito que demostrara Holland para los Algoritmos Genéticos se refiere al hecho de que mientras el algoritmo calcula las aptitudes de los individuos de una población, estima de forma implícita las aptitudes promedio de un número mucho más alto de cadenas cromosómicas a través del cálculo de las aptitudes promedio observadas en los “bloques constructores” que se detectan en la población.

<sup>4</sup> Un esquema es un patrón de valores de los genes de un cromosoma.

Friedberg [35] utilizó un algoritmo de asignación de crédito para dividir la influencia de diferentes instrucciones individuales en un programa. Este procedimiento fue comparado con una búsqueda puramente aleatoria, y en algunos casos fue superado por ésta. Tras ciertas modificaciones al procedimiento, Friedberg fue capaz de superar a una búsqueda totalmente aleatoria, pero no pudo resolver satisfactoriamente el problema de “estancamiento” (*stagnation*, en inglés) de la población que se le presentó y por ello fue cruelmente criticado por investigadores de la talla de Marvin Minsky, quien en un artículo de 1961 [51] indicó que el trabajo de Friedberg era “una falla total”. Minsky atribuyó el fracaso del método de Friedberg a lo que él denominó el “fenómeno de mesa” [52], según el cual el estancamiento de la población se debía al hecho de que sólo una instrucción del programa era modificada a la vez, y eso no permitía explorar una porción significativa del espacio de búsqueda. Aunque estas observaciones no son del todo precisas [26], el problema del estancamiento siguió siendo el principal inconveniente del procedimiento de Friedberg, aunque Fogel [26] considera que su trabajo precedió el uso de los sistemas de clasificadores que popularizara varios años después John Holland [41].

George J. Friedman [37] fue tal vez el primero en proponer una aplicación de técnicas evolutivas a la robótica: su tesis de maestría propuso evolucionar una serie de circuitos de control similares a lo que hoy conocemos como redes neuronales, usando lo que él denominaba “retroalimentación selectiva”, en un proceso análogo a la selección natural. Muchos consideran a este trabajo como el origen mismo de la denominada “robótica evolutiva”, que es una disciplina en la que se intentan aplicar técnicas evolutivas a diferentes aspectos de la robótica (planeación de movimientos, control, navegación, etc.). Desgraciadamente, las ideas de Friedman nunca se llevaron a la práctica, pero aparentemente fueron redescubiertas por algunos investigadores varios años después [26].

Los circuitos de control que utilizara Friedman en su trabajo modelaban a las neuronas humanas, y eran capaces de ser excitadas o inhibidas. Además, era posible agrupar estos circuitos simples (o neuronas) para formar circuitos más complejos. Lo interesante es que Friedman propuso un mecanismo para construir, probar y evaluar estos circuitos de forma automática, utilizando mutaciones aleatorias y un proceso de selección. Este es probablemente el primer trabajo en torno a lo que hoy se denomina “hardware evolutivo”.

Friedman [36] también especuló que la simulación del proceso de reproducción sexual (o cruce) y el de mutación nos conduciría al diseño de “máquinas pensantes”, remarcando específicamente que podrían diseñarse programas para jugar al ajedrez con este método.

Hans Joachim Bremermann [14] fue tal vez el primero en ver a la evolución como un proceso de optimización, además de realizar una de las primeras simulaciones de la evolución usando cadenas binarias que se procesaban por medio de reproducción (sexual o asexual), selección y mutación, en lo que sería otro claro predecesor del algoritmo genético. Bremermann [16] y Bremermann y Rogson [17] utilizaron una técnica evolutiva para problemas de optimización con restricciones lineales. La idea principal de su propuesta era usar un individuo factible el cual se modificaba a través de un operador de mutación hacia un conjunto de direcciones posibles de movimiento. Al extender esta técnica a problemas más complejos, Bremermann y Rogson utilizaron además operadores de recombinación especializados [18]. Bremermann fue uno de los

primeros en utilizar el concepto de “población” en la simulación de procesos evolutivos, además de intuir la importancia de la co-evolución [14] (es decir, el uso de dos poblaciones que evolucionan en paralelo y cuyas aptitudes están relacionadas entre sí) y visualizar el potencial de las técnicas evolutivas para entrenar redes neuronales [15].

Otros investigadores hicieron también importantes contribuciones para forjar lo que hoy se conoce como “computación evolutiva”, aunque en aras de la brevedad, no hablaremos sobre ellos en este artículo (los interesados, favor de consultar a Fogel en sus trabajos de 1995 [27], de 1998 [26] y a Ray en su trabajo de 1992 [56]).

## **4. Principales paradigmas de la Computación Evolutiva**

El término “computación evolutiva” o “algoritmos evolutivos” engloba una serie de técnicas inspiradas en los principios de la teoría Neo-Darwiniana de la evolución natural. En términos generales, para simular el proceso evolutivo en una computadora se requiere:

- Codificar las estructuras que se replicarán (o sea, una estructura de datos que se utilice para almacenar a un “individuo”).
- Operaciones que afecten a los “individuos” (típicamente, se usa cruce y mutación).
- Una función de aptitud que nos indique qué tan buena es una solución con respecto a las demás.
- Un mecanismo de selección que implemente el principio de “supervivencia del más apto” de la teoría de Darwin.

Aunque hoy en día es cada vez más difícil distinguir las diferencias entre los distintos tipos de algoritmos evolutivos existentes, por razones sobre todo históricas, suele hablarse de tres paradigmas principales:

- Programación Evolutiva
- Estrategias Evolutivas
- Algoritmos Genéticos

Cada uno de estos paradigmas se originó de manera independiente y con motivaciones muy distintas, por lo que procederemos a describir brevemente a cada uno de ellos de forma totalmente independiente.

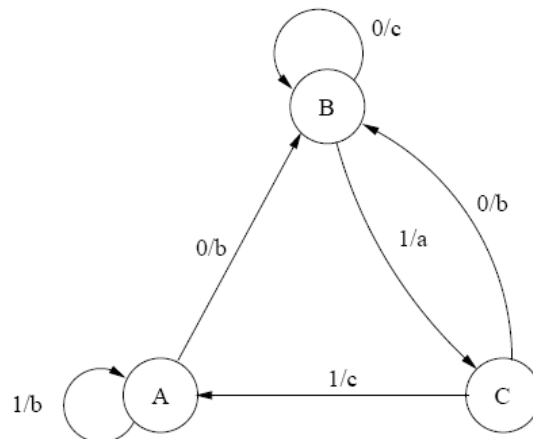


Figura 1: Autómata finito de 3 estados. Los símbolos a la izquierda de “/” son de entrada y los de la derecha son de salida. El estado inicial es C.

## 4.1. Programación Evolutiva

Lawrence J. Fogel propuso en los 1960s una técnica denominada “Programación Evolutiva”, en la cual la inteligencia se ve como un comportamiento adaptativo [29]. La Programación Evolutiva enfatiza los nexos de comportamiento entre padres e hijos, en vez de buscar emular operadores genéticos específicos (como en el caso de los Algoritmos Genéticos).

El algoritmo básico de la Programación Evolutiva es el siguiente:

- Generar aleatoriamente una población inicial.
- Se aplica mutación.
- Se calcula la aptitud de cada hijo y se usa un proceso de selección mediante torneo (normalmente estocástico) para determinar cuáles serán las soluciones que se retendrán.

La Programación Evolutiva es una abstracción de la evolución al nivel de las especies, por lo que no se requiere el uso de un operador de recombinación (diferentes especies no se pueden cruzar entre sí). Asimismo, usa selección probabilística. Resulta un tanto más fácil entender la Programación Evolutiva analizando un ejemplo como el que se muestra en la figura 1. La tabla de transiciones de este autómata es la siguiente:

Estado Actual	<b>C</b>	<b>B</b>	<b>C</b>	<b>A</b>	<b>A</b>	<b>B</b>
Símbolo de Entrada	0	1	1	1	0	0
Estado Siguiente	B	C	A	A	B	B
Símbolo de Salida	b	a	c	b	b	c

En este autómata pueden ahora aplicarse cinco diferentes tipos de mutaciones: cambiar un símbolo de salida, cambiar una transición, agregar un estado, borrar un estado y cambiar el

estado inicial. El objetivo es hacer que el autómata reconozca un cierto conjunto de entradas (o sea, una cierta expresión regular) sin equivocarse ni una sola vez.

Algunas aplicaciones de la Programación Evolutiva son [27]:

- Predicción
- Generalización
- Juegos
- Control automático
- Problema del viajero
- Planeación de rutas
- Diseño y entrenamiento de redes neuronales
- Reconocimiento de patrones

## 4.2. Estrategias Evolutivas

Las Estrategias Evolutivas fueron desarrolladas en 1964 en Alemania para resolver problemas hidrodinámicos de alto grado de complejidad por un grupo de estudiantes de ingeniería encabezado por Ingo Rechenberg [9].

La versión original (1+1)-EE usaba un solo padre y con él se generaba un solo hijo. Este hijo se mantenía si era mejor que el padre, o de lo contrario se eliminaba (a este tipo de selección se le llama *extintiva*, porque los peores individuos tienen una probabilidad cero de ser seleccionados).

En la (1+1)-EE, un individuo nuevo es generado usando:

$$\bar{x}^{t+1} = \bar{x}^t + N(0, \sigma)$$

donde  $t$  se refiere a la *generación* (o iteración) en la que nos encontramos, y  $N(0, \sigma)$  es un vector de números Gaussianos independientes con una media cero y desviación estándar  $\sigma$ .

Rechenberg [57] introdujo el concepto de población, al proponer una estrategia evolutiva llamada  $(\lambda + 1)$ -EE, en la cual hay  $\lambda$  padres y se genera un solo hijo, el cual puede reemplazar al peor padre de la población (selección extintiva).

Schwefel [60, 61] introdujo el uso de múltiples hijos en las denominadas  $(\mu + \lambda)$ -EEs y  $(\mu, \lambda)$ -EEs. La notación se refiere al mecanismo de selección utilizado:

- En el primer caso, los  $\mu$  mejores individuos obtenidos de la unión de padres e hijos sobreviven.
- En el segundo caso, sólo los  $\mu$  mejores hijos de la siguiente generación sobreviven.

Rechenberg [57] formuló una regla para ajustar la desviación estándar de forma determinista durante el proceso evolutivo de tal manera que el procedimiento convergiera hacia el óptimo. Esta se conoce como la “regla del éxito 1/5”, que indica que:



“La razón entre mutaciones exitosas y el total de mutaciones debe ser 1/5. Si es mayor, entonces debe incrementarse la desviación estándar. Si es menor, entonces debe decrementarse”.

Formalmente:

$$\sigma(t) = \begin{cases} \sigma(t-n)/c & \text{si } p_s > 1/5 \\ \sigma(t-n)*c & \text{si } p_s < 1/5 \\ \sigma(t-n) & \text{si } p_s = 1/5 \end{cases}$$

donde  $n$  es el número de dimensiones,  $t$  es la generación,  $p_s$  es la frecuencia relativa de mutaciones exitosas medida sobre intervalos de por ejemplo  $10n$  individuos, y  $c = 0.817$  (este valor fue derivado teóricamente por Schwefel [60]).  $\sigma(t)$  se ajusta cada  $n$  mutaciones.

Los operadores de recombinación de las Estrategias Evolutivas pueden ser:

- **Sexuales:** el operador actúa sobre 2 individuos elegidos aleatoriamente de la población de padres.
- **Panmíticos:** se elige un solo padre al azar, y se mantiene fijo mientras se elige al azar un segundo padre (de entre toda la población) para cada componente de sus vectores.

Algunas aplicaciones de las Estrategias Evolutivas son [60]:

- Problemas de rutas y redes
- Bioquímica
- Óptica
- Diseño en ingeniería
- Magnetismo

### 4.3. Algoritmos Genéticos

Los Algoritmos Genéticos (denominados originalmente “planes reproductivos genéticos”) fueron desarrollados por John H. Holland a principios de los 1960s [42, 43], motivado por su interés en resolver problemas de aprendizaje de máquina.

El algoritmo genético enfatiza la importancia de la cruce sexual (operador principal) sobre el de la mutación (operador secundario) y usa selección probabilística. El algoritmo básico es el siguiente:

- Generar (aleatoriamente) una población inicial.
- Calcular la aptitud de cada individuo.
- Seleccionar (probabilísticamente) con base a la aptitud.
- Aplicar operadores genéticos (cruza y mutación) para generar la siguiente población.
- Ciclar hasta que cierta condición se satisfaga.

La representación tradicional es la binaria, tal y como se ejemplifica en la figura 2. A la cadena binaria se le llama “cromosoma”. Al bloque de bits que codifica una sola variable del problema se le denomina “gen” y al valor dentro de cada posición cromosómica se le llama “alelo”.

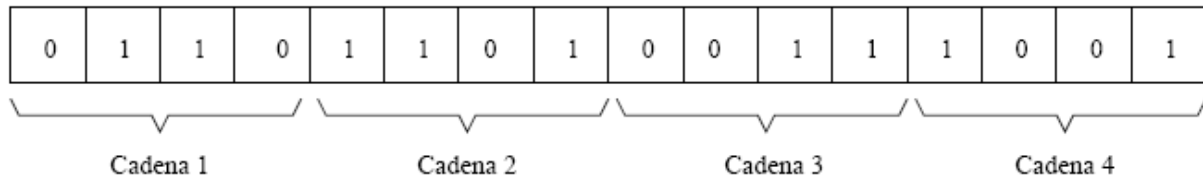


Figura 2: Ejemplo de la codificación (mediante cadenas binarias) usada tradicionalmente con los Algoritmos Genéticos.

Para poder aplicar el algoritmo genético se requiere de los 5 componentes básicos siguientes:

- Una representación de las soluciones potenciales del problema.
- Una forma de crear una población inicial de posibles soluciones (normalmente un proceso aleatorio).
- Una función de evaluación que clasifique las soluciones en términos de su “aptitud”.
- Operadores genéticos que alteren la composición de los hijos que se producirán para las siguientes generaciones (normalmente cruce sexual y mutación).
- Valores para los diferentes parámetros que utiliza el algoritmo genético (tamaño de la población, probabilidad de cruce, probabilidad de mutación, número máximo de generaciones, etc.)

Bajo representación binaria, suelen usarse 3 tipos principales de cruce: de un punto (ver figura 3), de dos puntos (ver figura 4) y uniforme (ver figura 5).

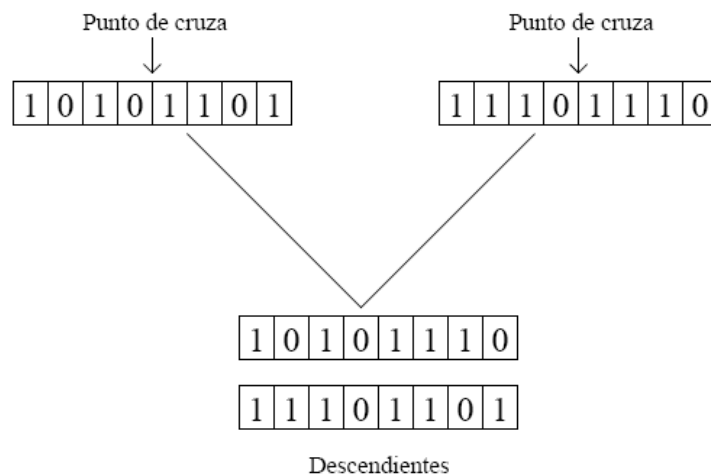


Figura 3: Cruce de un punto.

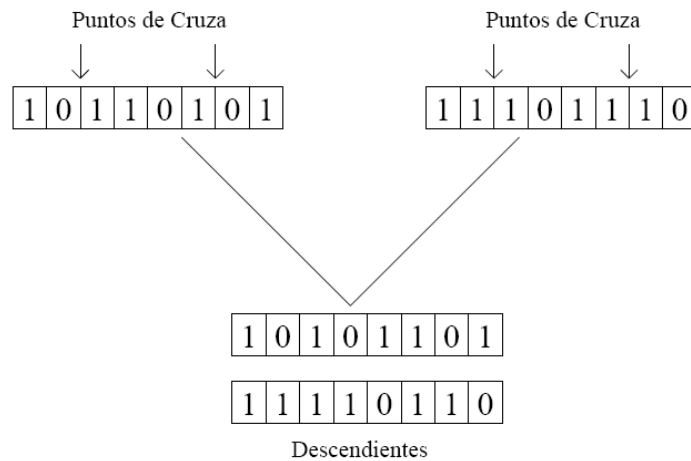


Figura 4: Cruce de dos puntos.

En la Figura 5 se observa la cruce uniforme con una probabilidad de 50%. Nótese que la mitad de los bits del Hijo 1 se tomaron del Padre 1 y la otra mitad, del Padre 2. El Hijo 2 obtiene sus bits de manera opuesta a como las obtiene el Hijo 1.

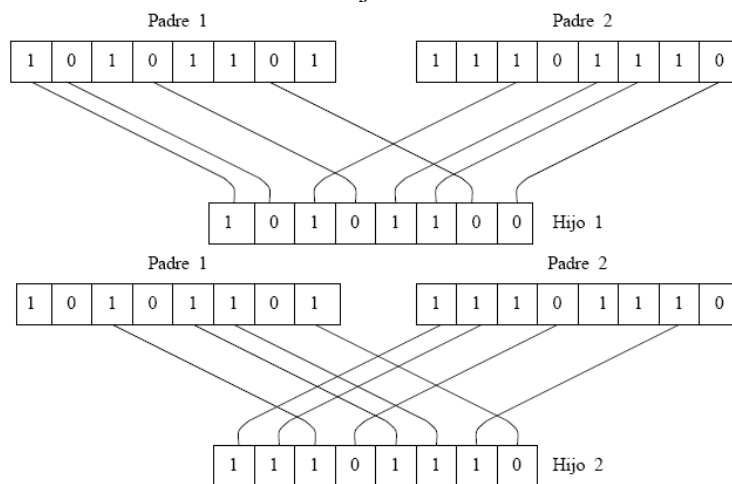


Figura 5: Cruce uniforme.

Algunas aplicaciones de los Algoritmos Genéticos son las siguientes [39]:

- Optimización (estructural, de topologías, numérica, combinatoria, etc.)
- Aprendizaje de máquina (sistemas clasificadores)
- Bases de datos (optimización de consultas)
- Reconocimiento de patrones (por ejemplo, imágenes o letras)
- Generación de gramáticas (regulares, libres de contexto, etc.)
- Planeación de movimientos de robots
- Predicción

## **5. Ventajas de las técnicas evolutivas**

Es importante destacar las diversas ventajas que presenta el uso de técnicas evolutivas para resolver problemas de búsqueda y optimización [39, 27]:

- Simplicidad Conceptual.
- Amplia aplicabilidad.
- Superiores a las técnicas tradicionales en muchos problemas del mundo real.
- Tienen el potencial para incorporar conocimiento sobre el dominio y para hibridarse con otras técnicas de búsqueda/optimización.
- Pueden explotar fácilmente las arquitecturas en paralelo.
- Son robustas a los cambios dinámicos.

Para finalizar, es importante mencionar que la computación evolutiva, como disciplina de estudio, ha atraído la atención de un número cada vez mayor de investigadores de todo el mundo. Esta popularidad se debe, en gran medida, al enorme éxito que han tenido los algoritmos evolutivos en la solución de problemas del mundo real de gran complejidad. De tal forma, es de esperarse que en los años siguientes el uso de este tipo de técnicas proliferará aún más. Nótese, sin embargo, que es importante tener en mente que los algoritmos evolutivos son técnicas heurísticas. Por tanto, no garantizan que convergerán al óptimo de un problema dado, aunque en la práctica suelen aproximar razonablemente bien el óptimo de un problema en un tiempo promedio considerablemente menor que los algoritmos deterministas. Esta distinción es importante, pues el papel de las técnicas heurísticas es el de servir normalmente como último recurso para resolver un problema en el que los algoritmos convencionales (típicamente deterministas) no funcionan o tienen un costo computacional prohibitivo. Esto implica que antes de decidir recurrir a los algoritmos evolutivos, debe analizarse la factibilidad de utilizar otro tipo de técnicas. Este paso, que pudiese parecer obvio para muchos, en la práctica suele omitirse en muchos casos y de ahí que exista bastante escepticismo por parte de aquellos que acostumbran a trabajar únicamente con algoritmos deterministas. El uso apropiado y pertinente de los algoritmos evolutivos será sin duda la base de su futuro como alternativa para la solución de problemas complejos y de ahí que se enfatice su importancia.

## **6. Aplicaciones de la Computación Evolutiva en Economía y Finanzas**

Aunque la computación evolutiva ha sido estudiada desde hace unos 40 años, su aplicación en el área de la Economía y las Finanzas es muy reciente, pues fue apenas en el año 1987 en el que comenzaron los primeros trabajos orientados a la Teoría de Juegos [8], para después crecer y extenderse al área de modelado macroeconómico [6, 48] y más recientemente en la ingeniería financiera [1, 10, 11]. A continuación se mencionan brevemente cada una de estas áreas y se discutirán los trabajos más importantes que se han desarrollado en cada una de ellas utilizando algoritmos evolutivos.

## 6.1. Teoría de Juegos

La primera aplicación de los Algoritmos Genéticos a las Ciencias Sociales fue un trabajo de Axelrod en el área de Teoría de Juegos que data de 1984 [7, 8]. Axelrod usó un Algoritmo Genético (AG) [39] para estudiar el Problema del Prisionero cuando se juega en repetidas ocasiones. Axelrod realizó un torneo entre diferentes computadoras, cada una de las cuales jugaba al Problema del Prisionero durante un número “n” de veces y en cada juego intenta jugar con la mejor estrategia para mejorar su ganancia. Se realizaban comparaciones de los resultados obtenidos por dos computadoras (jugadores), y con base en los resultados obtenidos se modificaba la estrategia para declararse culpable o inocente con cierta probabilidad, esto es, permitía que la estrategia fuese “evolucionando” con base en los resultados obtenidos. Su objetivo era encontrar una estrategia para los jugadores más robusta que la mejor conocida en aquel entonces para el Problema del Prisionero. Axelrod era un experto en Ciencias Políticas, pero estaba influenciado por el biólogo William Hamilton, quien utilizaba un modelo de competencia muy parecido a la selección natural (en la que se observa que una especie compete con otras en un nicho ecológico). John Miller, en su tesis doctoral de 1988 [50] (de la Universidad de Michigan), modeló también a un conjunto de jugadores como individuos en un algoritmo genético, los cuales respondían a estímulos de eventos externos. Este fue el inicio de la denominada “Escuela Michigan”, que iniciara y liderara por varios años John H. Holland. El trabajo pionero que realizara John Holland en los años 1960s y que luego continuara David Goldberg en los 1980s, tuvo una profunda influencia en diversas áreas del conocimiento, y la Economía no fue la excepción.

Otro trabajo pionero interesante ha sido el de Thomas Riechmann [58], que se ha enfocado a demostrar que la Teoría de Juegos puede arrojar luz sobre los fundamentos de los AGs. Motivado por el éxito que han tenido los AGs en la solución de problemas en Economía, Reichman se interesó en saber si habían ciertas propiedades en los AGs que fuesen las responsables de su éxito. Durante varios años, los teóricos de la computación evolutiva han usado cadenas de Markov para modelar el comportamiento de un algoritmo genético y para estudiar aspectos tales como su convergencia [23, 59]. El enfoque de Reichman, sin embargo, consiste en interpretar un AG como un juego de N-personas repetitivo. De tal forma, Riechmann [58] utiliza la estructura básica del algoritmo genético como modelo de aprendizaje en el que utiliza una población como algo más que una distribución de economías o un comportamiento en las estrategias a seguir por diferentes jugadores (o agentes). Así entonces, un agente económico intenta resolver un problema el cual utiliza una función que lidia directamente con el concepto del equilibrio de Nash. Una estrategia de Nash está definida como la mejor estrategia que pueden seguir los jugadores (o agentes); y es lo que todo agente económico busca encontrar al resolver cualquier problema, es decir, encontrar un equilibrio en el que se sienta satisfecho o ha encontrado la ganancia máxima del problema. Por tanto, Riechmann usa un algoritmo genético para modelar un sistema completo de agentes económicos, cada uno intentando llegar a un equilibrio de Nash. En términos económicos, esto significa que cada agente intenta coordinar una estrategia con los demás agentes, y la mejor forma de hacer esto es maximizando su beneficio (o utilidad o ganancia o todo lo que el modelo quiere que el agente maximice). Por tanto, una población de individuos genéticos se interpreta como una población de agentes, en la que cada uno está tratando de jugar una estrategia diferente, y los individuos más aptos son aquellos que obtienen un mayor beneficio.

Otro juego que se ha utilizado con algoritmos evolutivos, es el juego del ultimátum. Este problema ha sido un tema muy utilizado por economistas. En los últimos 20 años, varios laboratorios han experimentado con comportamientos humanos y éstos han conducido a la similitud de esos comportamientos con este juego. Este juego es interesante porque muestra la importancia de la imparcialidad, percibida subjetivamente por los jugadores y formando endógenos por medio de la interacción en sus estrategias en el trato final. John Duffy y Engle-Warnick [24] utilizan la Programación Matemática en vez de los análisis estadísticos comúnmente adoptados para inferir de datos experimentales las reglas de decisión que entes humanos podrían aplicar. Marimon [49] utiliza la regresión lineal para calcular las variables que cada humano puede usar para formar sus expectativas de ganancia.

La aplicación de Duffy y Feltovich [25] utiliza la programación genética para resolver el problema del juego del ultimátum, donde unos sujetos juegan el mismo juego durante 40 periodos. En el juego del ultimátum, dos jugadores, A y B, son denominados el proponente y el replicador, respectivamente, y éstos deben decidir cómo dividir un lote de \$10. El proponente propone una división del lote y el jugador B puede aceptar o rechazar la oferta o propuesta. Si se acepta, entonces cada quien se lleva la parte acordada del lote y si la oferta es rechazada, entonces ninguno se lleva nada del lote.

## **6.2. Uso de agentes artificiales para la Bolsa de Valores**

En esta área es donde más trabajos se han publicado en los últimos años y donde se muestra el mayor potencial para el uso de los algoritmos evolutivos. En 1988, cuando John Holland y Brian Arthur establecieron el programa de Economía en el Instituto Santa Fe, el problema de simulación de mercados o mercados artificiales se escogió como el punto de partida para este proyecto de investigación.

Yang [66] siguiendo el trabajo de Chan [20] utilizó una Red Neuronal para un problema de doble subasta en el que desarrolló una estrategia que los agentes o jugadores deben seguir para maximizar las ganancias en un mercado artificial. Este mercado artificial lo modeló con una red neuronal la cual reducía la línea existente entre los mercados simulados y los reales. El mercado simulado de Yang es capaz de tener un comportamiento parecido a un equilibrio de esperanzas homogéneas racionales (*homogeneous rational expectations equilibrium*) en el que los consumidores de un mercado simulado se comportan igual de racionales que los consumidores de un mercado real.

Shu-Heng Chen [21], propuso un software llamado AIE-ASM, el cual realiza simulaciones de acciones en bolsa basadas en algunos modelos de precios activos. Una característica distinguible de este software es que el comportamiento de los agentes es modelado utilizando la programación genética en vez de los Algoritmos Genéticos. La programación genética es realmente una variante del algoritmo genético en la cual los individuos son representados mediante árboles (o programas de computadora), en vez de usar cadenas binarias, como en los Algoritmos Genéticos. La principal motivación de este trabajo radica en la reproducibilidad de los diferentes trabajos que se hacen en cuanto al tema de modelos económicos, pues estos trabajos carecen de la información necesaria para reproducir los mismos resultados de los

experimentos que cada uno reporta en sus artículos. De tal forma, este software aborda el problema de las acciones en bolsa simulando los agentes artificialmente. La programación genética se utiliza para guiar el aprendizaje dinámico de los varios comerciantes, que son los que toman las decisiones de compra y venta en una bolsa.

Jasmina Arifovic [4], fue la primer persona en obtener un doctorado con una tesis relacionada con aplicaciones de los Algoritmos Genéticos en macroeconomía. Arifovic publicó también varios artículos en el tema de intercambio de divisas [2, 3]. En estos estudios, utilizó los Algoritmos Genéticos para entrenar agentes simulados en varios modelos con dos diferentes monedas. Implementó los Algoritmos Genéticos en el método de aprendizaje y consideró varias alternativas posibles, como por ejemplo optimizar el aprendizaje y utilizar el conocimiento adquirido para realizar predicciones en este tipo de problemas de intercambio de divisas. En estos problemas se pretende optimizar las ganancias obtenidas de la toma de decisiones al elegir un portafolio de inversiones adecuado.

También existen aplicaciones en las que se utilizan los algoritmos evolutivos para abordar problemas de microeconomía como el control de la contaminación en un ambiente de simulación de problemas económicos, y el problema de encontrar el rol adecuado de las instituciones en el ámbito social; estos trabajos se describen brevemente a continuación.

Bell y Beare [12] se ocupan del problema del control de la contaminación ambiental. En esta propuesta se utiliza un algoritmo genético para emular una subasta mercantil de emisiones contaminantes. El proceso se modeló utilizando un juego evolutivo no cooperativo, en el que los agentes aprenden la mejor estrategia de apuesta (que es independiente de las demás estrategias). En particular, el valor de un “permiso” (así se le llama al agente) depende de ambos factores: precio y cantidad de emisiones contaminantes que se venden a otros participantes en el juego. Los resultados en las simulaciones muestran que un simple intercambio de emisiones resulta más efectivo cuando el número de participantes es pequeño. Cuanto más grande es el número de participantes, el desempeño disminuye. Los resultados indican que un simple impuesto es una medida más efectiva que un mecanismo de mercado para evitar una mayor contaminación en el ambiente.

Francesco Luna [47] se concentra en el problema del rol de las instituciones en un proceso de aprendizaje en agentes económicos. En este trabajo se utiliza una red neuronal para modelar el proceso de aprendizaje, y el algoritmo genético se usa para desarrollar y obtener los pesos en esta red de la mejor forma posible para optimizar el aprendizaje.

David Fogel [28] realizó un estudio sobre la representación de los agentes inteligentes que se suelen utilizar en problemas económicos. Fogel indica que existen clases de modelos que se usan para representar a los agentes inteligentes, y demostró que estos modelos afectan considerablemente el desempeño del mismo. Este estudio lo valida en dos diferentes problemas muy utilizados por los economistas como lo son “el problema del Farol” [5] y el problema del Prisionero [55]. También demuestra que el uso de diferentes ajustes en los parámetros de simulación como cambiar de una respuesta discreta a una continua, o el incluir un proceso inteligente a los agentes (jugadores) es suficiente para obtener un comportamiento muy diferente en los resultados.

### 6.3. Ingeniería Financiera

La primera aplicación de los algoritmos evolutivos en econometría fue desarrollada por John Koza [45]. En ella se dice que un problema importante en Economía es encontrar una relación matemática entre las variables observadas empíricamente para medirlas en un sistema. En las técnicas tradicionales de modelado, uno necesita escoger el tamaño y la forma del modelo para después encontrar los valores de algunos coeficientes necesarios para utilizar un modelo en particular y poder obtener el mejor ajuste entre los datos observados y el modelo real. Para Koza, el encontrar la forma de un modelo puede ser vista como el buscar en un espacio de búsqueda conformado por ciertos programas de computadora que produzcan una salida para ciertas entradas. Koza hizo uso de la programación genética para formular una ley fundamental en Economía llamada “la ecuación de intercambio o la teoría cuántica del dinero”.

George Szpiro [63], propone el uso de Algoritmos Genéticos para buscar dependencias entre conjuntos de datos para problemas en Finanzas y Economía. Se usan los algoritmos genéticos para problemas en minería de datos. Éstos se muestran idóneos para detectar e identificar relaciones ocultas en espacios multi-dimensionales que son caracterizados por la existencia de múltiples óptimos. De tal forma, se deja que los Algoritmos Genéticos descubran ecuaciones matemáticas que recrean, o al menos, imitan las fórmulas reales de generación de datos.

En [54], Robert Pereira muestra como es que la computación evolutiva resulta relevante en el estudio de las estrategias comerciales. Comienza con la revisión de dos clases básicas de reglas comerciales, llamadas reglas de moving-average (MA) y reglas de order-statistics (OS). Luego, utiliza los Algoritmos Genéticos para encontrar los parámetros óptimos que se utilizan en estas dos últimas reglas comerciales. En este trabajo, se utilizaron datos reales del mercado Australiano comprendidos del 4 de Enero de 1982 al 31 de Diciembre de 1989. Las reglas óptimas que se encontraron, se utilizaron después para evaluarlas y probar su habilidad de pronosticar el beneficio económico durante el periodo comprendido entre el 2 de Enero de 1990 al 31 de Diciembre de 1997. Los resultados indicaron que las reglas encontradas por el algoritmo genético mejoraron las estrategias seguidas por un simple mecanismo de “ajuste de riesgo”.

Raymond Tsang y Paul Lajbcygier [64] ofrecen una alternativa al uso de los Algoritmos Genéticos estándar. Ellos proponen dividir el algoritmo genético en un modelo de islas, en el que la población de soluciones se divide en un número fijo de subpoblaciones, cada una evolucionando de forma independiente, pero todas resolviendo el mismo problema. Así pues, proponen el uso de un operador de migración el cual periódicamente intercambia soluciones entre las subpoblaciones. Adoptando este tipo de Algoritmos Genéticos, se pueden diferenciar las subpoblaciones con diferentes parámetros de control y la búsqueda se hace más extensa y mucho más efectiva. Las simulaciones en problemas de estrategias comerciales evidencian una superioridad de los Algoritmos Genéticos con esquemas de islas sobre los Algoritmos Genéticos tradicionales.

Existe un artículo de Nemara Chidambaran [22] en el que éste lidia con el problema de la elección de precios con Programación Genética (PG). Primero se examina cómo es que la programación genética mejora el modelo de “Black-Scholes” bajo ciertas condiciones, y después



utiliza este modelo para resolver un problema con datos del mundo real. En el estudio de la simulación que se hace, muestra que las fórmulas de la programación genética obtienen mejores resultados que el modelo de Black-Scholes en los diez casos de estudio que proponen. En problemas de índices S&P, la propuesta de programación genética gana en 9 de 10 diferentes casos. El modelo de Black-Scholes pretende medir variación de los instrumentos financieros en el tiempo, puede usarse para determinar el precio de algún producto en un tiempo determinado. Este modelo es uno de los más importantes en la teoría financiera moderna. Desarrollado por Fisher Black y Myron Scholes en 1973 [13] como una de las mejores opciones para determinar precios justos.

Otra aplicación relacionada con el problema de elección de precios es en la que se involucra la *volatilidad*. Medir la volatilidad financiera es uno de los problemas más difíciles en las Finanzas económicas de la actualidad. La volatilidad implica que para cierto problema de elección de precios se tiene que invertir la función de Black-Scholes, pero esto no es una tarea sencilla y se realiza con varias aproximaciones analíticas para atacar este problema como lo es el método de Newton-Raphson. Christian Keber [44] muestra que la programación genética puede usarse para derivar aproximaciones muy precisas y determinar la volatilidad en este tipo de problemas. Keber genera 1.000 soluciones aleatorias y con base en éstas deriva una expresión analítica utilizando programación genética. La aproximación obtenida se prueba en dos conjuntos de datos tomados en la literatura y en dos más generados artificialmente. Para realizar las comparaciones, se utiliza el modelo de Black-Scholes como el modelo a vencer. Los resultados muestran que las fórmulas dadas por la aproximación evolutiva resultan más precisas en los problemas con más parámetros.

## 6.4. Otros desarrollos actuales

La Computación Evolutiva se ha convertido en una metodología eficiente para resolver problemas financieros y económicos. Ha demostrado ser una herramienta muy poderosa en problemas que los métodos analíticos no logran resolver. Como resultado, existen varias conferencias y talleres internacionales que intentan cubrir este tópico. Algunos ejemplos son las sesiones especiales en computación evolutiva en Finanzas y Economía que se organizan en el *IEEE Congress on Evolutionary Computation* (CEC), la *Conference on Computational Intelligence in Economics and Finance* (CIEF); y el *Workshop on Economic Heterogeneous Interacting Agents* (WEHIA).

Hasta ahora, la computación evolutiva ha sido aplicada en problemas de pronóstico financiero, estimación de parámetros econométricos, modelos macroeconómicos, elección de precios, simulación de mercados financieros y procesos sociales. Como sabemos, los problemas del mundo real involucran una alta complejidad, ambientes con ruido, imprecisiones e incertidumbre. Por esta razón, el uso de las técnicas evolutivas se vuelve necesario para problemas en diversas áreas en Finanzas y Economía.

En cuanto a publicaciones recientes sobre uso de algoritmos evolutivos en Finanzas y Economía, algunos de ellos se detallan a continuación:

- Streichert, Ulmer y Zell en 2004 [62] realizó un estudio comparativo de distintos tipos de cruce en un algoritmo evolutivo con representación real para resolver problemas de portafolios de inversión basados en el modelo de media-varianza de Markowitz. Este problema trata de elegir un conjunto de activos que tienen un cierto precio, y se busca maximizar las ganancias obtenidas de elegir los activos correctos.
- Asunción Mochón y co-autores en 2005 [53] hicieron un análisis del uso de la computación evolutiva en problemas de subastas. Se diseñó una simulación para examinar el comportamiento de las subastas, buscando encontrar la estrategia óptima de los apostadores para problemas dinámicos de subastas. El algoritmo propuesto se probó en varios experimentos con muchos apostadores y muchas subastas simultáneas, y los resultados indicaron que el algoritmo evolutivo es capaz de encontrar estrategias que mejoran a las estrategias canónicas que usualmente se utilizan.
- García-Almanza y Tsang en 2006 [38] presenta un método para simplificar los árboles de decisión con ayuda de la programación genética. En él se identifican y se eliminan las reglas que causan errores en la clasificación y, como consecuencia, se obtiene un árbol más eficiente para tomar decisiones en problemas de mercados financieros.
- Jin Li en 2006 [46] presenta un algoritmo que utiliza la programación genética para resolver problemas multi-objetivo para pronosticar sistemas financieros. En este caso, se resuelven problemas de pronósticos financieros en los que se hace uso de la dominancia de Pareto para encontrar varias soluciones eficientes en una misma ejecución, permitiendo al usuario tomar una decisión más completa.

## 7. Conclusiones

En este artículo hemos proporcionado una introducción general a la computación evolutiva, incluyendo su desarrollo histórico y una descripción breve de los tres paradigmas principales que la conforman. En la parte final del artículo se describen diversas aplicaciones de los algoritmos evolutivos en la economía y las finanzas, incluyendo varios trabajos pioneros en el área.

A lo largo del artículo, se ha buscado despertar el interés de los especialistas en economía y finanzas por el uso de los algoritmos evolutivos. La motivación principal para usar estas técnicas radica en su amplia aplicabilidad (pueden usarse para problemas de optimización no lineales de enorme complejidad) y su facilidad de uso. Las aplicaciones aquí reportadas sirven como evidencia tangible de dichas ventajas.

Finalmente, se hace notar que a pesar de existir ya diversos trabajos importantes en economía y finanzas, existen todavía muchas áreas de oportunidad para nuevos desarrollos. Por ejemplo, el uso de algoritmos evolutivos para estudiar patrones de consumo, para esquemas de asignación de crédito o para predecir tendencias de las acciones en la bolsa de valores, son sólo algunas de las muchas áreas de oportunidad que todavía existen para los interesados en trabajar en este apasionante campo.

## Referencias

- [1] F. Allen and F. Karjalainen, *Using Genetic Algorithms to Find Technical Trading Rules*, White Center for Financial Research, The Wharton School, 1993, pp. 20 - 93.
- [2] J. Arifovic, *The Behavior of the Exchange Rate in the Genetic Algorithm and Experimental Economics.*, *Journal of Political Economy*, 104 (1996), pp. 510-541.
- [3] J. Arifovic, *Evolutionary Dynamics of Currency Substitution*, *Journal of Economic Dynamics and Control*, 25 (2001), pp. 395-417.
- [4] J. Arifovic, *Learning by Genetic Algorithms in Economic Environments*, University of Chicago, Chicago, 1991.
- [5] W. B. Arthur, *Effective Choice in the Iterated Prisoner's Dilemma*, *Journal Conflict Resolution*, 24 (1994), pp. 3-25.
- [6] W. B. Arthur., *On Learning and Adaptation in the Economy*, Sante FI Economics Research Program Working, 1992.
- [7] R. Axelrod, *The Evolution of Cooperation*, Basic Books, New York, 1984.
- [8] R. Axelrod, *The Evolution of Strategies in the Iterated Prisoner's Dilemma*, in D. L., ed., *Genetic Algorithms and Simulated Annealing*, Pittmann, London, 1987, pp. 32 - 41.
- [9] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.
- [10] R. J. Bauer, *Genetic Algorithms and Investment Strategies*, Wiley, New York, 1994.
- [11] R. J. Bauer and G. E. Liepins, *Genetic Algorithms and Computerized Trading Strategies*, in D. E. O'leary and R. R. Watkins, eds., *Expert Systems in Finance*, North Holland, 1992.
- [12] R. Bell and S. Beare, *Emuating Trade in Emissions Permits: An Application of Genetic Algorithms*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 159 - 174.
- [13] F. Black and M. Scholes, *The Pricing of Options and Corporate Liabilities*, *Journal of Political Economy*, 81 (1973), pp. 637 - 654.

- [14] H. J. Bremermann, *The evolution of intelligence. The nervous system as a model of its environment.*, University of Washington, Seattle, 1958.
- [15] H. J. Bremermann, *Numerical optimization procedures derived from biological evolution processes.*, in H. L. Oestreicher and D. R. Moore, eds., *Cybernetic Problems in Bionics*, Gordon and Breach, New York, 1968, pp. 543 - 561.
- [16] H. J. Bremermann, *Optimization through evolution and recombination.*, in M. C. Yovits, G. T. Jacobi and G. D. Goldstein, eds., *Self-Organizing Systems*, Spartan Books, Washington D.C., 1962, pp. 93 - 106.
- [17] H. J. Bremermann and M. Rogson, *An evolution-type search method for convex sets.*, ONR, Berkeley, California, 1964.
- [18] H. J. Bremermann and M. Rogson, *Global properties of evolution processes.*, in H. H. Pattee, E. A. Edlsack, L. Fein and A. B. Callahan, eds., *Natural Automata and Useful Simulations*, Spartan Books, Washington D.C., 1966, pp. 3 - 41.
- [19] W. D. Cannon, *The Wisdom of the Body*, Norton and Company, New York, 1932.
- [20] N. T. Chan, B. LeBaron and A. W. Poggio, *Information Dissemination and Aggregation in Asset Markets with Simple Intelligent Traders*, AIM-1646, 1998.
- [21] S.-H. Chen, C.-H. Yeh and C.-C. Liao, *On AIE-ASM: Software to Simulate Artificial Stock Markets with Genetic Programming*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 107 - 122.
- [22] N. Chidambaran, J. Triqueros and C.-W. J. Lee, *Option Pricing Via Genetic Programming*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 383 - 397.
- [23] H. Dawid, *Adaptive Learning by Genetic Algorithms*, Springer, Berlin, 1996.
- [24] J. Duffy and J. Engle-Warnick, *Using Symbolic Regression to Infer Strategies*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 61 - 84.
- [25] J. Duffy and N. Feltovich, *Observation of Others Affect Learning in Strategic Environments? An Experimental Study*, *International Journal of Game Theory*, 28 (1999), pp. 131-152.
- [26] D. B. Fogel, *Evolutionary Computation. The Fossil Record. Selected Readings on the History of Evolutionary Algorithms.*, New York, 1998.

- [27] D. B. Fogel, *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*, The Institute of Electrical and Electronic Engineers, New York, 1995.
- [28] D. B. Fogel, K. Chellapilla and P. Angeline, *Evolutionary Computation and Economic Models: Sensitivity and Unintended Consequences*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 245 - 269.
- [29] L. J. Fogel, A. J. Owens and M. J. Walsh, *Artificial Intelligence through Simulated Evolution.*, John Wiley & Sons, Inc., New York, 1966.
- [30] A. S. Fraser, *The evolution of purposive behavior*, in H. von Foerster, J. D. White, L. J. Peterson and J. K. Russell, eds., *Purposive Systems*, Spartan Books, Washington D.C., 1968, pp. 15 - 23.
- [31] A. S. Fraser, *Simulation of Genetic Systems by Automatic Digital Computers I. Introduction*, Australian Journal of Biological Sciences, 10 (1957), pp. 484 - 491.
- [32] A. S. Fraser, *Simulation of Genetic Systems by Automatic Digital Computers II. Effects of Linkage on Rates of Advance Under Selection.*, Australian Journal of Biological Sciences, 10 (1957), pp. 150 - 162.
- [33] A. S. Fraser, *Simulation of Genetic Systems by Automatic Digital Computers VI. Epistasis*, Australian Journal of Biological Sciences, 13 (1960), pp. 150 - 162.
- [34] A. S. Fraser and D. Burnell, *Computer Models in Genetics*, Mc. Graw Hill, New York, 1970.
- [35] R. M. Friedberg, *A Learning Machine: Part I*, IBM Journal of Research and Development, 2 (1958), pp. 2 - 13.
- [36] G. J. Friedman, *Digital simulation of an evolutionary process*, General Systems: Yearbook of the Society for General Systems Research, 4 (1959), pp. 171 - 184.
- [37] G. J. Friedman, *Selective Feedback Computers for Engineering Synthesis and Nervous System Analogy*, University of California, Los Angeles, 1956.
- [38] A. L. García-Almanza and E. Tsang, *Simplifying Decision Trees Learned by Genetic Programming*, *IEEE Congress on Evolutionary Computation (CEC 2006)*, Vancouver, Canada, 2006, pp. 7906-7912.
- [39] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

- [40] A. Hoffmann, *Arguments on Evolution: A Paleontologist's Perspective*, Oxford University Press, New York, 1989.
- [41] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Harbor: University of Michigan Press, 1975.
- [42] J. H. Holland, *Concerning efficient adaptive systems*, in M. C. Yovits, G. T. Jacobi and G. D. Goldstein, eds., *Self-Organizing Systems—1962*, Spartan Books, Washington D.C., 1962, pp. 215 - 230.
- [43] J. H. Holland, *Outline for a logical theory of adaptive systems*, *Journal of the Association for Computing Machinery*, 9 (1962), pp. 297 - 314.
- [44] C. Keber, *Evolutionary Computatioin in Option Pricing: Determining Implied Volatilities Based on American Put Options*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 399 - 415.
- [45] J. Koza, *A Genetic Approach to Econometric Modelling*, in P. Bourguine and B. Walliser, eds., *Economics and Cognitive Science*, Pergamon Press, 1992, pp. 57-75.
- [46] J. Li, *Enhancing Financial Decision Making Using Multi-Objective Financial Genetic Programming*, *IEEE Congress on Evolutionary Computation (CEC 2006)*, Vancouver, Canada, 2006, pp. 7935-7942.
- [47] F. Luna, *Computable Learning, Neural Networks and Institutions*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 211 - 232.
- [48] R. Marimon, E. McGrattan and T. J. Sargent, *Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents.*, *Journal of Economic Dynamics and Control*, 14 (1990), pp. 329-373.
- [49] R. Marimon, S. E. Spear and S. Sunder, *Expectationally Driven Market Volatility: An Experimental Study*, *Journal of Economic Theory*, 61 (1993), pp. 74-103.
- [50] J. H. Miller, *Two Essays on the Economics of Imperfect Information*, *University of Michigan, PhD Thesis*, 1988.
- [51] M. L. Minsky, *Steps toward artificial intelligence*, *Proceedings of the IRE*, 49 (1961), pp. 8 - 30.
- [52] M. L. Minsky and O. G. Selfridge, *Learning in random nets*, in C. Cherry, ed., *Proceedings of the 4th London Symposium on Information Theory*, London, 1961.

- [53] A. Mochón, D. Quintana, Y. Sáez and P. Isasi, *Analysis of Ausubel Auctions by Means of Evolutionary Computation*, *Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, Scotland, 2005, pp. 2645-2652.
- [54] R. Pereira, *Forecasting Ability But No Profitability: An Empirical Evaluation of Genetic Algorithm-Optimised Technical Trading Rules*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 287 - 309.
- [55] A. Rapoport, *Optimal Policies for the Prisoner's Dilemma*, Univ. North Carolina, Chapel Hill, 1966.
- [56] T. S. Ray, *An approach to the synthesis of life*, in C. G. Langton, C. Taylor, J. D. Farmer and S. Rasmussen, eds., *Artificial Life II*, Addison-Wesley, Reading, Massachusetts, 1992, pp. 371 - 408.
- [57] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart, Alemania, 1973.
- [58] T. Riechmann, *Genetic Algorithm Learning and Economic Evolution*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, Heidelberg New York, 2002, pp. 45 - 59.
- [59] G. Rudolph, *Convergence Properties of Canonical Genetic Algorithms*, *IEEE Transaction on Neural Networks*, 5(1) (1994), pp. 96-101.
- [60] H.-P. Schwefel, *Numerical Optimization of Computer Models*, Wiley, Chichester, UK., 1981.
- [61] H.-P. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, Basel, Alemania, 1977.
- [62] F. Streichert, H. Ulmer and A. Zell, *Evaluating a Hybrid Encoding and Three Crossover Operators on the Constrained Portfolio Selection Problem*, *Congress on Evolutionary Computation (CEC 2004)*, Portland, Oregon, 2004, pp. 932-939.
- [63] G. G. Szpiro, *A Search for Hidden Relationships: Data Mining with Genetic Algorithms*, *Computational Economics*, 10 (3) (1997), pp. 267-277.
- [64] R. Tsang and P. Lajbcygier, *Optimizing Technical Trading Strategies with Split Search Genetic Algorithms*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, Physica-Verlag, New York, 2002, pp. 333 - 357.

- [65] A. M. Turing, *Computing Machinery and Intelligence*, *Mind*, 59 (1950), pp. 94 - 101.
- [66] J. Yang, *The Efficiency of an Artificial Double Auction Stock Market with Neural Learning Agents*, in S.-H. Chen, ed., *Evolutionary Computation in Economics and Finance*, physica - verlag, New York, 2002, pp. 85 - 105.