

SISTEMAS INTELIGENTES APLICADOS A REDES DE DATOS

Leonardo César Corbalán

Director: Ing. Armando de Giusti - Codirector: Lic. Laura Lanzarini

*Facultad de Informática
Universidad Nacional de La Plata*

Carrera de de Especialidad en Redes

Noviembre de 2006

INDICE

1	INTRODUCCIÓN	1
2	DISEÑO ÓPTIMO DE REDES CONFIABLES.....	2
2.1	ACERCA DEL DISEÑO DE REDES CONFIABLES	2
2.2	SOLUCIONES CON ALGORITMOS EVOLUTIVOS.....	4
3	RUTEO EN REDES DE DATOS	10
3.1	ACERCA DEL RUTEO EN REDES DE DATOS	10
3.2	SOLUCIONES CON COLONIA DE HORMIGAS	11
4	TRANSMISIÓN MULTICAST	15
4.1	ACERCA DE LA TRANSMISIÓN MULTICAST.....	15
4.2	SOLUCIONES CON ALGORITMOS EVOLUTIVOS.....	17
5	SISTEMAS DE DETECCIÓN DE INTRUSOS.....	18
5.1	ACERCA DE LOS SISTEMAS DE DETECCIÓN DE INTRUSOS	18
5.2	SOLUCIONES CON REDES NEURONALES ARTIFICIALES.....	20
5.3	SOLUCIONES CON SISTEMAS INMUNES ARTIFICIALES	25
6	RECUPERACIÓN DE LA INFORMACIÓN EN INTERNET	27
6.1	ACERCA DE LA RECUPERACIÓN DE LA INFORMACIÓN EN INTERNET.....	27
6.2	SOLUCIONES CON AGENTES INTELIGENTES.....	28
7	CONCLUSIONES.....	30
8	REFERENCIAS	32

1 Introducción

El presente trabajo monográfico tiene por objetivo presentar una visión general sobre el estado actual y la actividad científica relacionada con los sistemas inteligentes aplicados en áreas inherentes a las redes de datos.

He recopilado una vasta cantidad de publicaciones vinculadas con la temática aquí pretendida. Sin embargo, la enumeración detallada de éstas resultaría tediosa y de escasa utilidad. En su lugar he estructurado la presentación del material agrupándolos en cinco tópicos generales con los que espero haber realizado una buena cobertura de los temas abordados.

Pretendo no profundizar en detalles que oscurezcan la visión general que deseo brindar. Sin embargo, cada temática estará acompañada de las referencias correspondientes para que el lector interesado pueda hallar fácilmente las fuentes originales y ampliar los conceptos de su interés.

A medida que vaya avanzando en los temas propuestos, y conforme sea necesario, iré introduciendo de forma muy reducida los conceptos más generales de las estrategias inteligentes utilizadas en cada caso.

Este trabajo está organizado de la siguiente manera. En la sección 2 abordaré la temática del diseño óptimo de redes confiables y algunas soluciones aportadas por los Algoritmos Evolutivos. La sección 3 tratará el tema de ruteo en redes y la aplicación de los sistemas basados en Colonias de Hormigas. La problemática sobre cómo lograr una transmisión multicast eficiente será el tema de la sección 4 presentando algunas soluciones evolutivas para resolverlo. En la sección 5 presentaré una importante área de aplicación de las Redes Neuronales y los Sistemas Inmunes Artificiales en el contexto de la seguridad informática: los Sistemas de Detección de Intrusos. Finalmente, antes de las conclusiones, dedicaré la sección 6 al tratamiento de la recuperación de la información en Internet utilizando Agentes Inteligentes.

2 Diseño Óptimo de Redes Confiables

2.1 Acerca del Diseño de Redes Confiables

El diseño incorrecto de una red de información puede impactar con consecuencias indeseables sobre el funcionamiento de los sistemas que la utilizan. Al hablar de diseño me estoy refiriendo especialmente a la forma de conectar los nodos de la red -topología- aunque también se podrían incluir otros aspectos como por ejemplo el tipo de enlace entre cada par de nodos. La topología está íntimamente ligada a las facilidades, posibilidades y restricciones de los servicios que corren sobre la red. Es por ello que resulta de sumo interés en el área contar con las herramientas y técnicas necesarias para poder llevar a cabo el diseño de una red en la forma más conveniente posible.

Dada la ubicación de los nodos, encontrar cuáles deben interconectarse, respetando un conjunto de restricciones, es un problema no trivial. Generalmente consiste en hallar la mejor topología de la red que maximice las prestaciones y minimice los costos. Este análisis suele suponer redes con y sin tolerancia a fallas. La tolerancia a fallas es otra de las restricciones impuestas que hay que conservar y que se logra generalmente con redundancia, por ejemplo garantizando más de una ruta posible entre cualquier par de nodos.

La optimización del diseño de redes con restricciones tiene un amplio rango de aplicación. Es particularmente importante en las redes de conmutación de circuitos a nivel de backbone, interconexión de redes, redes de fibra óptica para telefonía, computadoras tolerantes a fallas y arquitectura de switches. [Magnago 2006].

Los problemas de diseño de topologías de redes sujeto al cumplimiento de determinadas restricciones -minimizar costos, garantizar conectividad, minimizar retardos etc.- están comprendidos en una clase de problemas más general conocida con el nombre de optimización combinatoria. Estos últimos pertenecen al conjunto de problemas NP considerados por la comunidad científica como intratables con el modelo de computadora vigente en la actualidad.

Si bien no hay certeza teórica sobre la no existencia de algoritmos eficientes de complejidad temporal polinómica (pertenecientes a P) para la resolución

de los problemas NP (no ha sido demostrado formalmente que $P \neq NP$) lo cierto es que no se ha encontrado ninguno. Este asunto ha inquietado a los teóricos de la computación por varias décadas y aunque está fuera de los alcances del presente trabajo, el lector interesado puede consultar [Hopcroft 1993] [Garey 2000] para un estudio detallado del mismo.

Resumiendo lo antes expuesto la situación es la siguiente: es necesario obtener una buena topología que cumpla con determinadas restricciones para lograr un diseño eficiente y confiable de la red. Sin embargo, los teóricos de la computación aseguran que actualmente no existe y posiblemente nunca exista, un algoritmo que resuelva este problema en un tiempo razonable.

No obstante, existen soluciones plausibles aportadas por una importante disciplina de las Ciencias de la Computación, la Inteligencia Artificial. Esa misma disciplina que fuera vislumbrada por el matemático inglés Alan Turing al que hoy le debemos no sólo el conocimiento de los principios fundamentales de la Teoría de la Computación sino también los primeros cuestionamientos que condujeron al nacimiento de la Inteligencia Artificial. Si bien algunos autores sitúan este acontecimiento en la reunión celebrada en el Dartmouth College (Hanover, EEUU) en 1956, donde se acuñó el nombre de Inteligencia Artificial, otros aseguran que esta disciplina nació a partir del artículo titulado "Computing Machinery and Intelligence" publicado por Turing en 1950 que comienza con la famosa frase "*Propongo considerar la siguiente cuestión: ¿Pueden las máquinas pensar?*". Además Turing propuso el único método hasta ahora conocido para determinar si las máquinas podrían tener la capacidad de pensar, actualmente conocido como test de Turing.

Hallar un buen diseño de la topología de una red con restricciones a cumplir, utilizando las herramientas con que cuenta la Inteligencia Artificial, es perfectamente posible. Si bien la construcción de un algoritmo que analice todas las posibles soluciones encontrando la óptima resulta prohibitivo por las razones antes descritas, la Inteligencia Artificial, al igual que la inteligencia humana, es capaz de ofrecer un conjunto de soluciones "razonables". Dichas soluciones, aunque pueden no ser óptimas, muchas veces están muy cerca de serlo y todo ello se logra en tiempos realmente aceptables. Para lograrlo suelen utilizarse búsquedas basadas en heurísticas.

Entre las metaheurísticas¹ más utilizadas en esta área de problemas se encuentran los *Algoritmos Evolutivos* (AE) y las *Colonias de Hormigas* (ACO por Ant-Colony Optimization). Aplazaré el comentario sobre los sistemas ACO hasta el tratamiento del ruteo en redes de datos, de esta forma pretendo no recargar esta sección con descripciones de sistemas inteligentes. Por el contrario, intentaré introducirlos en forma gradual a medida que vaya desarrollando los tópicos de interés para este trabajo monográfico centrado en las tecnologías de redes de datos.

2.2 Soluciones con Algoritmos Evolutivos

Los Algoritmos Evolutivos son uno de los representantes más importantes de la inteligencia artificial subsimbólica². También forman parte de un conjunto de técnicas que se reconocen como bio-inspiradas. Constituyen un proceso de búsqueda estocástico basado en la teoría de selección natural de Darwin. El espíritu de esta teoría está sintetizado en la conocida frase "*supervivencia del más apto*". Así, conforme se sucedan las generaciones en una población de individuos, los genes que determinen características ventajosas en los organismos que lo posean tenderán a prevalecer sobre los otros. Aquellos descendientes que por mutaciones o entrecruzamiento genético permitan obtener una nueva característica beneficiosa tendrán mayores posibilidades de transferir sus "buenos genes" a las siguientes generaciones pues son quienes potencialmente tendrán mayor cantidad de descendientes.

Los Algoritmos Evolutivos recrean la teoría darwiniana para resolver problemas de búsqueda y optimización. Trabajan con una población de potenciales soluciones al problema a resolver representadas en una estructura de datos que recibe el nombre de cromosoma. A cada solución se le asigna un valor de aptitud, también conocido como fitness, que cuantifica qué tan bien resuelve el problema en cuestión.

¹ Tanto los Algoritmos Evolutivos como las Colonias de Hormigas son considerados metaheurísticas. A diferencia de las heurísticas que son estrategias siempre específicas para resolver un problema particular, las metaheurísticas constituyen métodos genéricos, aplicables a distintos problemas que guían las operaciones de los métodos heurísticos subordinados para producir eficientemente soluciones de alta calidad.

² El enfoque subsimbólico de la Inteligencia Artificial se caracteriza por crear sistemas con capacidad de aprendizaje. Éste se puede obtener a nivel de individuo imitando el cerebro –Redes Neuronales–, de colonia imitando el comportamiento social de las hormigas -Ant Colony Optimization- o a nivel de especie, imitando la evolución -Computación Evolutiva-. En todos los casos se parte de representaciones de bajo nivel –neuronas, insectos y cromosomas– que se manipulan para conseguir el objetivo propuesto –de alto nivel de abstracción–, por eso se dice que, a diferencia de la Inteligencia Artificial Simbólica, el enfoque es de abajo hacia arriba (Bottom-Up).

Cuanto mayor sea este valor mayor será la probabilidad de que la misma sea seleccionada para reproducirse, cruzando su material genético -cromosoma- con el de otra solución. Este cruce producirá nuevas soluciones -descendientes de las anteriores- las cuales comparten algunas de las características de sus padres. Cuanto menor sea el fitness de la solución, menor será la probabilidad de elección para la reproducción, y por tanto tendrá menor oportunidad de que su material genético se propague en sucesivas generaciones³.

Así, a lo largo de las generaciones, las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados -mejores soluciones-, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el Algoritmo Evolutivo ha sido bien diseñado, la población convergerá, idealmente, hacia una solución óptima del problema.

El poder de los Algoritmos Evolutivos proviene del hecho de trabajar sobre el espacio de representación -genotipo-, en lugar de hacerlo con el espacio de las soluciones -fenotipo-. Esto le permite una gran independencia del problema.

Los Algoritmos Evolutivos procesan similitudes en el genotipo junto con información sobre las capacidades de supervivencia en el entorno actual del fenotipo correspondiente. Al explotar una información tan fácilmente disponible, estos algoritmos se pueden aplicar en prácticamente cualquier problema. Este hecho es también el responsable, según Goldberg [Goldberg 1989], de su eficiencia, ya que al operar en el nivel de código genético, son difíciles de engañar aun cuando la función que codifican los cromosomas sea difícil para los enfoques tradicionales.

Aunque Goldberg se refería a una clase específica de Algoritmo Evolutivo denominada Algoritmos Genéticos, su apreciación es válida también en el contexto de cualquier estrategia evolutiva. Por razones históricas los algoritmos genéticos ocupan un lugar de privilegio en esta área de la Inteligencia Artificial subsimbólica. Ello se debe al trabajo de John Holland quien a fines de los '60 desarrolló los planes evolutivos que se popularizaran con el nombre de Algoritmos Genéticos en el libro "*Adaptation in Natural and Artificial Systems*" publicado en 1975. Los algoritmos

³ Cabría esperar que un individuo mal adaptado no tenga ninguna chance de reproducirse. Esta sería una decisión desacertada, ya que ciertas particularidades de un individuo mal adaptado combinado con otros podría ser muy útil, quizá la llave para salir de una solución subóptima podría estar en algún segmento de un individuo con bajo fitness.

genéticos utilizan representación binaria en los cromosomas y proponen al crossover -entrecruzamiento- como operador genético principal. A diferencia de estos últimos, los Algoritmos Evolutivos, no imponen restricción alguna sobre el tipo de representación u operadores genéticos utilizados.

La anatomía de un algoritmo evolutivo simple puede verse en la figura 1.

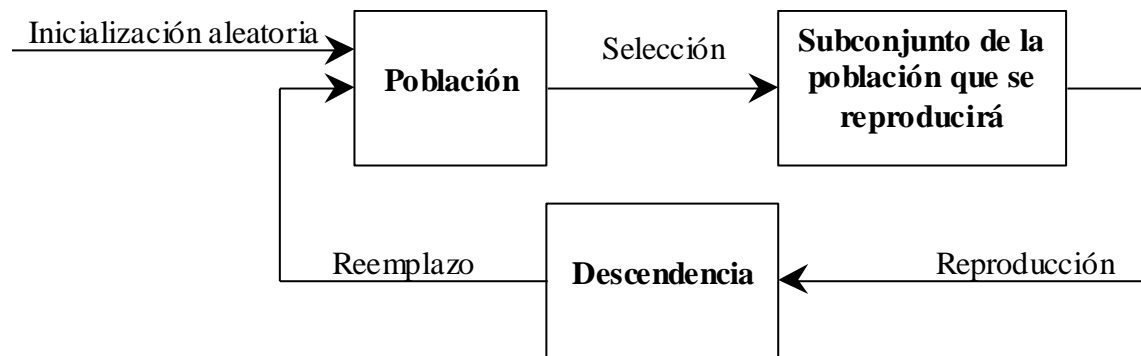


Figura 1. Esquema de un algoritmo evolutivo

Por lo general se comienza con una población de cromosomas generados al azar y se itera el bucle de *selección*, *reproducción* y *reemplazo* hasta que se cumpla alguna condición de finalización, pudiendo ser esta un número máximo de generaciones, cuando el algoritmo ha convergido, es decir, un gran porcentaje de la población tiene características similares, o cuando el algoritmo no logra mejores soluciones durante una determinada cantidad de generaciones.

La selección se realiza en base a la función de fitness que califica las potenciales soluciones para que, aquellas con valores más altos, tengan mayor probabilidad de reproducirse. Los operadores genéticos más utilizados en la reproducción son el crossover y la mutación. El primero consiste en crear la descendencia por recombinación del material genético de los padres, el segundo, que se aplica con mucha menos probabilidad, se realiza cambiando al azar el valor de un gen en el cromosoma. Por último, existen varios métodos de reemplazo para determinar quienes serán sustituidos por la nueva descendencia lograda para conformar la siguiente generación.

El lector interesado puede consultar el libro "*Genetic Algorithms in Search, Optimization, and Machine Learning*" de David Goldberg, discípulo de Holland, en dónde se tratan distintas estrategias para el diseño de algoritmos genéticos. Debo advertir, sin embargo, que la superioridad de esta clase de algoritmos en relación

con otras estrategias evolutivas y su sustentación matemática han sido posteriormente desacreditadas. En su tesis doctoral Cota Porras [CottaPorras 1998] resume una serie de argumentos esgrimidos por varios autores en esa dirección. Por esta razón en la actualidad se prefiere el concepto más general de Algoritmos Evolutivos que no impone restricción alguna en la representación cromosómica. Un autor muy referenciado en la comunidad de Computación Evolutiva es Zbigniew Michalewicz quien posee varios libros publicados en el tema.

Volviendo al problema del diseño de redes óptimas basado en restricciones describiré en líneas generales cuál es la forma de abordarlo utilizando alguna estrategia evolutiva.

Podemos modelar una red de información por medio de un grafo formado por nodos y enlaces que denominaremos links. A cada link se le asocia algún valor que representa el costo o peso de esa conexión. En esencia se trata de algún parámetro que caracteriza el enlace y que interviene directa o indirectamente en el criterio de optimización. En algunos casos se pueden considerar simultáneamente más de un parámetro lo que convierte el problema en una optimización multiobjetivo. La utilización de estrategias evolutivas aplicadas en este contexto dan lugar a lo que se conoce como Optimización Evolutiva Multiobjetivo⁴.

El problema de hallar el mejor diseño de la red se reduce entonces a encontrar el mejor grafo que garantice las restricciones impuestas. Para resolverlo con alguna estrategia evolutiva es necesario crear la población de cromosomas que representen las potenciales soluciones buscadas. Para ello se debe concebir alguna forma de representar un grafo en un cromosoma.

Una representación muy utilizada por los investigadores es la matricial. La topología de una red con n nodos a interconectar queda especificada por una matriz $A_{n \times n}$ donde $A(i,j)$ representa el parámetro de interés que caracteriza el link que une el nodo i con el nodo j . Un valor especial puede significar la no existencia de conexión. Si el enlace es bidireccional, lo que sucede frecuentemente, alcanza con codificar en el cromosoma los elementos $A(i,j)$ tales que $i < j$.

Una vez establecida la forma de codificar los grafos como cromosomas estamos en condiciones de utilizar alguna estrategia evolutiva conocida para resolver el problema de optimización. En ocasiones es posible ensayar alguna variante intentando aprovechar el conocimiento de alguna particularidad del problema en cuestión. Por ejemplo pueden enriquecerse los genes de la población con soluciones buenas previamente conocidas. Algunos investigadores han propuesto utilizar para ello soluciones obtenidas con otras heurísticas.

Aún en este punto, el diseñador del algoritmo debe lidiar con algunas particularidades del problema. Por ejemplo, es posible que no todas las conexiones sean admitidas en una topología factible de implementarse, lo que conduce a considerar algunas soluciones como inválidas. En general existen distintas estrategias para tratar con este inconveniente. Una forma consiste en utilizar el concepto de *penalización* asignando un valor de fitness nulo a los cromosomas que generan fenotipos inválidos. De esta manera, el algoritmo evolutivo irá eliminando estos genes de la población conforme avancen las generaciones. Otra estrategia se basa en la utilización de alguna representación que asegure la imposibilidad de mantener soluciones inválidas, por ejemplo codificando en los cromosomas sólo los enlaces permitidos. En ocasiones suelen utilizarse operadores de crossover *ad-hoc* que garanticen no producir estos genotipos indeseables. Muy extendidas también están las estrategias de reparación que consisten en transformar las soluciones inválidas en las válidas más cercanas utilizando algún criterio de distancia.

Existen muchas otras cuestiones interesantes para discutir sobre la aplicación de Algoritmos Evolutivos en esta área de problemas. Sin embargo, profundizar en más detalles puede oscurecer la visión general sobre cómo los sistemas inteligentes pueden contribuir con buenas soluciones en el área de redes de datos, que es el objetivo de esta monografía.

Quienes estén interesados en la discusión detallada de algunos de los temas expuestos en este apartado pueden encontrar una gran cantidad de artículos

⁴ En el sitio web <http://www.jeo.org/emo> puede consultarse un enorme repositorio de referencias bibliográficas con tesis doctorales, artículos de revistas y congresos internacionales en el área de Optimización Evolutiva Multiobjetivo mantenido por el Dr. Carlos A. Coello Coello

publicados en el área. Sólo a modo de ejemplo enumeraré algunos trabajos de investigación en los siguientes párrafos.

Dengiz y Fulya [Dengiz 1997a] [Dengiz 1997b] implementaron un Algoritmo Evolutivo para resolver el problema del diseño de redes confiables sobre un enorme espacio de búsqueda con buenos resultados.

Reichelt, Rothlauf y Gmilkowsky [Reichelt 2004] utilizaron heurísticas de reparación de redes no factibles para resolver el diseño óptimo de redes confiables. Los autores exponen que las funciones de penalización no son apropiadas para resolver el problema y que el uso de heurísticas de reparación basadas en el número de *spanning trees*⁵ en la red obtienen siempre mejores soluciones con menor esfuerzo computacional.

Konak y Smith [Konak 2005] propusieron una solución evolutiva para el diseño de redes confiables utilizando como criterio de diseño una medida de resistencia de la red para asegurar la integridad de los servicios de red en caso de falla de algún componente.

Magnago [Magnago 2006] en su tesis de magister de redes de datos (2006) realiza un estudio sobre el Diseño Óptimo de Redes de mínimo costo, sujeto a restricciones de confiabilidad con el supuesto de fallas en los enlaces y propone como estrategia de resolución un algoritmo evolutivo con operadores genéticos *ad-hoc* enriquecidos con conocimiento específico del problema, lo cual permite acotar el espacio de búsqueda del algoritmo. En la actualidad existe dentro de la comunidad de inteligencia artificial una línea de investigación sobre las estrategias para introducir información específica del problema a resolver en los Algoritmos Evolutivos.

Finalmente, la optimización de grafos es una técnica sumamente útil que puede utilizarse también en otros problemas inherentes a las redes de datos. La asignación óptima de tareas en un sistema distribuido es un ejemplo de ello. Se trata de encontrar la mejor distribución de las tareas en un conjunto de procesadores conociendo los requerimientos de comunicación entre ellas y sus respectivos costos de procesamiento. El problema se modela con un grafo sobre el que deben encontrarse los

⁵ Se dice que un árbol T es un Spanning Tree de un grafo G conectado, si T es un subgrafo de G y contiene todos los nodos de G.

conjuntos de corte que cumplan mejor los objetivos propuestos: minimizar costos de comunicación y tiempos de retorno, maximizar el paralelismo y la utilización eficiente de los recursos, etc. [Ardenghi 2005]. Este planteo representa un nuevo problema de optimización combinatoria.

3 Ruteo en Redes de Datos

3.1 Acerca del Ruteo en Redes de Datos

En una red de área local (LAN), por lo general no es necesario que el emisor localice al receptor antes de realizar una transmisión. Simplemente coloca el dato en la red para que el receptor pueda recibirlo. No obstante una red de área amplia (WAN) consta de un gran número de máquinas, cada una de ellas con cierto número de líneas hacia otras máquinas. Para que un mensaje llegue del emisor al receptor, tiene que hacer un cierto número de saltos y, en cada uno de ellos, elegir una línea a utilizar. El proceso relacionado con la elección de la mejor ruta se llama ruteo y es la tarea principal de la capa de red -la capa 3 del modelo OSI desarrollado por la International Standards Organization (ISO)- [Tanenbaum1995]

En ocasiones, la mejor ruta no es la más corta, sino la menos congestionada con la que se minimiza el retraso en la transmisión de información. La cantidad de tráfico de datos en las conexiones suele ser un factor dinámico que puede variar significativamente. Así los retardos en las transmisiones también se verán afectados con el curso del tiempo. Algunos algoritmos de ruteo intentan adaptarse a cargas cambiantes, mientras que otros se limitan a tomar decisiones con base en promedios a largo plazo.

Para poder lograr un ruteo eficiente es necesario contar con un conjunto de métricas que nos permitan caracterizar los enlaces. Estas métricas están asociadas a distintas magnitudes como ser distancia, costo, retardo, número de saltos *-hops-* etc.

En general podemos clasificar a los algoritmos de ruteo existentes en dos categorías: estáticos y adaptativos o dinámicos. Los estáticos en general son más fáciles de implementar pero menos eficientes que los adaptativos.

En los algoritmos de ruteo estático, los ruteadores toman decisiones basándose en una configuración previamente fijada por el administrador de la red sin tener en cuenta lo que está ocurriendo en cada instante de tiempo. Por el contrario, los dinámicos o adaptativos basan sus decisiones en consideraciones sobre el estado actual de la red. Estos algoritmos permiten adecuarse a las condiciones cambiantes y por ello consiguen un mejor rendimiento promedio que los estáticos. Por ejemplo si un enlace se cae, el ruteador reaccionará inmediatamente cambiando las rutas para que no se intente utilizar el enlace que está fuera de servicio.

Los algoritmos dinámicos son siempre los preferidos. Los más utilizados son *vector-distancia* y *estado de enlace*. En el algoritmo vector-distancia los ruteadores se comunican con sus vecinos intercambiando la información sobre las distancias estimadas hacia los nodos de toda la red. Así cada ruteador recalcula sus estimaciones cada vez que recibe esta información de sus vecinos. RIP es un protocolo muy popular que implementa este algoritmo. En el algoritmo estado de enlace un ruteador comunica a los restantes nodos de la red cuáles son sus vecinos y a qué distancias está de ellos. Con la información que los nodos de la red reciben de todos los demás se construye un "mapa" de la red y sobre él se calculan los caminos óptimos. OSPF es un protocolo que implementa este algoritmo.

Puede hallarse una descripción detallada de los conceptos de ruteo sobre redes TCP/IP en [Comer 1995] capítulos 14, 15 y 16.

Los sistemas inteligentes más aplicados en esta área de problemas son los conocidos como *Colonias de Hormigas* o simplemente sistemas ACO (Ant Colony Optimizacion). Las propuestas evolutivas son casi inexistentes. Sin embargo debo destacar el trabajo de Masaharu Munetomo [Munetomo 1998] [Munetomo 1999] quien propone un algoritmo genético para ruteo adaptativo en Internet llamado GARA (Genetic Adaptive Routing Algorithm)

3.2 Soluciones con Colonia de Hormigas

Comentaré brevemente el funcionamiento y la metáfora subyacentes en estos sistemas inteligentes. Las estrategias ACO están inspiradas en el estudio del comportamiento de las hormigas sociales, realizado principalmente por Pierre Paul

Grassé y Edgard O. Wilson. Estos investigadores entienden a la colonia como el organismo, y a las hormigas como sus componentes u órganos.

Pierre Paul Grassé introdujo el concepto de *estigmergia* para explicar cómo se lograban realizar las tareas en insectos sociales sin necesidad de planificación ni de un poder central. Estigmergia significa colaboración a través del medio físico.

En sistemas descentralizados, tales como las colonias de hormigas, los diferentes componentes colaboran a través de pautas o hitos dejados en el medio como por ejemplo feromonas, acumulación de objetos o cualquier otro tipo de cambio físico, como la temperatura. De esta forma se puede explicar la realización de obras colectivas sin necesidad de la intervención de una autoridad central.

El comportamiento de las hormigas en el rastro que dejan de feromonas y su seguimiento, ha sido estudiado por varios investigadores por medio de experimentos controlados. Uno de éstos, consiste en un puente doble que conecta un nido de hormigas y una fuente de comida. La longitud de uno de los caminos es el doble del otro. Después de algún tiempo, se observa que las hormigas eligen el camino más corto.

Ello se explica de la siguiente manera: Cuando las hormigas caminan hacia la fuente de comida y de regreso a su nido van depositando sobre la tierra una sustancia llamada feromona. Las hormigas tienden a seguir este rastro de feromonas eligiendo con alta probabilidad el camino marcado con fuertes concentraciones de esta sustancia. Cuando las hormigas llegan por primera vez al puente doble tienen que decidir cual camino tomar. Al no haber aún ninguna cantidad de feromona depositada en ninguno de los dos senderos eligen cualquiera aleatoriamente. Las hormigas que eligieron el camino más corto, llegan más rápido a la fuente de comida y comienzan su regreso al nido; esto hace que se acumule mayor cantidad de feromonas en el puente más corto que en el largo, y a pesar de que hay hormigas que se van por el camino más largo, con el paso del tiempo, todas utilizan el puente más corto (figura 2).

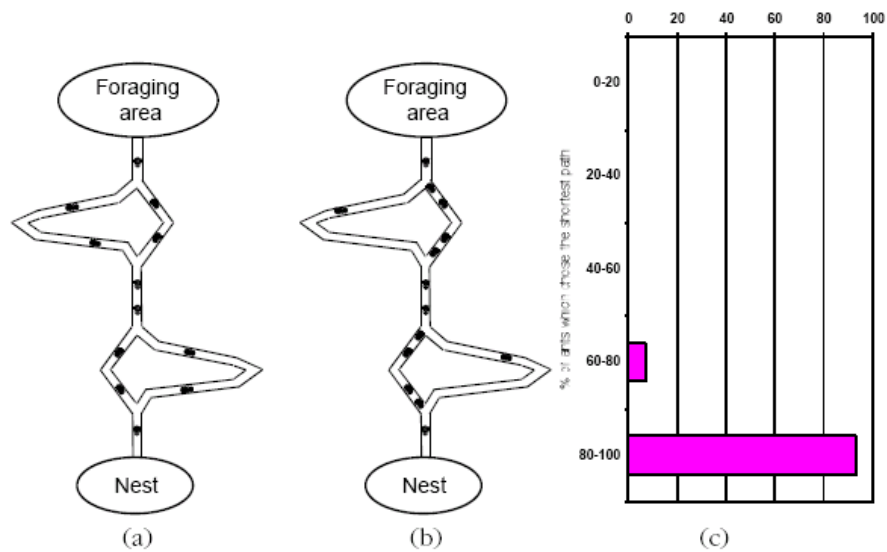


Figura 2. Experimento puente doble. (a) Las hormigas comienzan la exploración del puente doble. (b) Eventualmente la mayoría de las hormigas elige el camino más corto. (c) Distribución del porcentaje de hormigas que eligen el camino más corto. Imagen extraída de [Dorigo 99]

Marco Dorigo fue quien propuso el primer algoritmo inspirado en el comportamiento de las hormigas para resolver problemas de optimización combinatoria en un reporte técnico en 1991 [Dorigo1991] y en su tesis doctoral en 1992. [Dorigo1992]

Se puede decir que los algoritmos ACO implementan un conjunto de agentes concurrentes -las hormigas- que se mueven aplicando políticas de decisión locales estocásticas a través de un espacio de estados correspondientes a soluciones parciales del problema a resolver. En estos sistemas es necesario simular la acción de las hormigas que van depositando feromonas pero también el efecto de evaporación de estas partículas que ocurre como consecuencia del transcurso del tiempo.

A diferencia de lo que ocurre con los Algoritmos Evolutivos, hay abundante cantidad de referencias a trabajos de investigación que utilizan esta metaheurística para resolver problemas relacionados con el ruteo en redes de información. Entre las soluciones existentes se destaca particularmente AntNet, un algoritmo de ruteo adaptativo propuesto por Dorigo y Di Caro en 1997. Una descripción detallada del método puede encontrarse en el reporte técnico [Dorigo 1997]. Al menos dos implementaciones distintas de AntNet están descritas en [Di Caro 1998]. También otros investigadores han propuesto variantes que mejoran el rendimiento del algoritmo

en algún aspecto. A modo de ejemplo mencionaré el trabajo de Benjamín Barán y Rubén Sosa que puede consultarse en [Barán 2001].

Otros investigadores han utilizado el concepto de estigmergia de una manera un poco distinta a la utilizada por las hormigas reales, y lo han hecho para reducir el overhead propio de los algoritmos de ruteo vector-distancia como el utilizado por el protocolo RIP. Recordemos que por estigmergia se entiende colaboración a través del medio y constituye una forma de comunicación indirecta entre agentes autónomos. Las hormigas utilizan el rastro de feromona como un estímulo positivo que las atrae induciéndolas a proseguir por el mismo camino. En [Amin 2003] Amin y Mikler presentan un algoritmo que utiliza la feromona como estímulo negativo que repele a los agentes para obtener una buena dispersión de los mismos en una red de comunicación.

La estrategia presentada por Amin y Mikler intenta disminuir el overhead ocasionado por la necesidad de comunicar el estado de las tablas de ruteo en los algoritmos vector-distancia. Para ello, en lugar de los típicos intercambios de mensajes de actualización entre nodos adyacentes, proponen utilizar un conjunto de agentes -hormigas- que recorren la red llevando información de ruteo. Así, la sobrecarga para mantener actualizadas las rutas en todos los nodos queda limitada a la cantidad de agentes en el sistema. Como estrategia para recorrer los nodos de la red las hormigas, en su trayectoria, van depositando en los links cierta cantidad de una feromona especial que repele en lugar de atraer a las demás hormigas. Así se obtiene una buena dispersión de los agentes en toda la red con resultados que superan ampliamente a la estrategia conocida como *random walk*.

Otro aspecto muy interesante del trabajo de estos investigadores es el control autónomo del tamaño de la colonia en lo que han denominado "*dynamic agent population*". Para ello se valen nuevamente de un tipo especial de feromona que van depositando en los nodos visitados. Recordemos que en todos estos sistemas es necesario simular la evaporación de las feromonas conforme pasa el tiempo. Así un nodo que tenga una gran concentración de esta sustancia indicará que ha sido muy visitado últimamente. Por el contrario un nivel muy bajo revela escasa actividad reciente de las hormigas en el sitio.

La cantidad de agentes en el sistema se regula de forma autónoma de la siguiente manera: Cuando una hormiga alcanza un determinado nodo percibe la cantidad de feromona que aún permanece en el sitio. Si su concentración es muy baja, por debajo de un determinado *umbral de clonación*, la hormiga se clona agregando un nuevo individuo al sistema que comenzará a recorrerlo desde ese lugar. Si por el contrario la cantidad de la substancia es excesiva, mayor que un *umbral de terminación*, el agente se autoelimina contribuyendo a disminuir la actividad relacionada con el intercambio de rutas en esa área.

Para finalizar este apartado comentaré que existe más investigación en el tema y que pueden encontrarse otros artículos que presentan soluciones a problemas de ruteo utilizando esta metaheurística como es el caso de [Mykkeltveit 2004] y [Jha 2005].

A modo de comentario indicaré que los sistemas ACO también se han utilizado en otras cuestiones relativas a las redes de datos como ser la búsqueda distribuida de fallas, la localización de servicios en redes Ad-Hoc y la búsqueda de recursos en redes P2P [Dan 2004].

4 Transmisión Multicast

4.1 Acerca de la Transmisión Multicast

El Multicast -o multidifusión- consiste en la transmisión simultánea de datos desde un proceso fuente a un grupo de procesos destino que residen en un subconjunto de nodos de la red de computadoras. El objetivo del multicast es que cada proceso perteneciente a un grupo reciba los mensajes enviados al mismo, frecuentemente con garantías de que han sido entregados. Estos sistemas de comunicación son extremadamente sofisticados; incluso la multidifusión IP, que proporciona garantías de entrega mínimas, requiere grandes esfuerzos de realización. Como preocupaciones principales están la eficiencia en el tiempo consumido y en el uso de ancho de banda, que suponen un reto incluso para grupos estáticos de procesos. Estos problemas se multiplican cuando los procesos pueden unirse o dejar los grupos de forma arbitraria [Colouris 2001].

La característica esencial de la comunicación por multicast es que un proceso realiza una única operación para enviar un mensaje a cada uno de los miembros de un grupo en lugar de realizar múltiples envíos sobre los procesos individuales. Esto implica mucho más que una ventaja para el programador ya que permite desarrollar una implementación eficiente la cual redundará en garantías de entrega más fuertes que las que serían posible obtener de otra forma.

En particular, la transmisión por multicast hará mucho más eficiente la utilización del ancho de banda. La implementación puede proceder de tal forma que sólo envíe el mensaje una vez sobre cada enlace de comunicación, basándose en un *árbol de distribución* -spanning tree-. Para ver estas ventajas imaginemos una transmisión de video donde un profesor explica un tema particular en una clase a distancia. Supongamos que dicha clase está siendo recibida por 20 nodos en una misma red local Ethernet ubicada en otro continente. Si no se utiliza multicast, el emisor enviará 20 copias de cada mensaje de forma independiente provocando demoras y requerimientos de ancho de banda. En cambio, si se emplea multicast, un conjunto de ruteadores intermedios dirigirán el tráfico de una única copia del mensaje desde el emisor hasta un ruteador en la LAN de destino. Este ruteador final utiliza entonces hardware de multicast (proporcionado por la Ethernet) para entregar de una sola vez el mensaje a los destinatarios en lugar de tener que enviarlos a los 20 nodos por separado.

Es precisamente por la aparición de este tipo de nuevas aplicaciones punto a multipunto en redes de datos como las transmisiones de radio y TV, video bajo demanda, teleconferencias y aprendizaje a distancia, que recientemente se ha incrementado el interés en los algoritmos de ruteo multicast. Dichas aplicaciones generalmente requieren de algunos parámetros de calidad de servicio -QoS por *Quality of Service*- tales como retardo máximo de extremo a extremo y recursos mínimos de ancho de banda.

También pueden considerarse otros parámetros importantes para implementar un sistema de multicast eficiente como por ejemplo mantener un buen balance de carga y un adecuado uso de los recursos de la red. El enfoque tradicional utilizado para el balance de la carga es la minimización de la utilización del enlace más sobrecargado, o minimización de la utilización máxima de los enlaces de la red.

4.2 Soluciones con Algoritmos Evolutivos

En general todas las soluciones planteadas en el área giran alrededor de la obtención del árbol de distribución multicast más adecuado que garantice las restricciones impuestas. Esto constituye un nuevo problema de optimización de grafos como los que se han descrito al tratar el tema "Diseño Optimo de Redes Confiables" en las primeras páginas de esta monografía. Por lo tanto, también representan un problema de optimización combinatoria. Los Algoritmos Evolutivos han mostrado sobrada evidencia de su capacidad para conseguir buenos resultados en este tipo de problema. Varios investigadores del área han realizado distintas propuestas evolutivas para resolverlos.

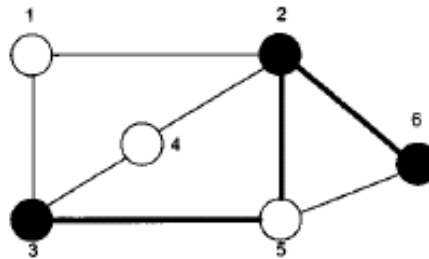


Figura 3. Modelo de una red con un grupo multicast en el que se ha remarcado un árbol de distribución multicast, una posible solución a codificar en un cromosoma. Imagen extraída de [Hamdam 2004]

Existen varios trabajos en el área que utilizan Algoritmos Evolutivos para resolver el problema del ruteo multicast sujeto a restricciones, intentando por ejemplo minimizar el retardo máximo de extremo a extremo. Las soluciones propuestas varían en cuanto a la representación de un árbol de distribución multicast como un cromosoma, a la aplicación de distintos operadores genéticos y a los mecanismos de selección y reparación empleados.

En [Ravikumar 1998] se propone un método basado en un algoritmo genético simple, en el cual la función de fitness está dada como la inversa del costo del árbol multicast y penalizada cuando el retardo máximo sobrepasa la cota superior prefijada. Uno de los principales inconvenientes que tiene el establecimiento a priori de esta cota de retardo máximo es que se pueden descartar soluciones de bajo costo con un retardo apenas superior al máximo permitido. Otros investigadores han propuesto distintas mejoras a este trabajo [de Arauo 2002].

En [Karabi 2004] se presenta otro método evolutivo para resolver el ruteo multicast con restricción de mínimo costo. Los autores proponen una nueva forma de codificación cromosómica acompañada de una función de reparación que corrige los genotipos inválidos. A diferencia del enfoque convencional que representa dentro del cromosoma una potencial solución al problema -árbol de distribución multicast- esta propuesta busca codificar un grafo genérico.

En [Hamdam 2004] se presenta una solución evolutiva considerando dos restricciones de calidad de servicio que la red debe proveer: retraso máximo extremo a extremo (end-to-end) y variación máxima del retraso de la transmisión por los distintos caminos desde la fuente a cada uno de los destinos.

Existe una gran actividad en la comunidad de investigadores del área. Se han propuesto diversas estrategias evolutivas que en general se diferencian por la forma en que codifican los árboles -soluciones- en los cromosomas, cuáles son y cómo se aplican los operadores genéticos, cómo se tratan las soluciones inválidas, si la representación y operadores elegidos son capaces de generarlas, y cuáles son los métodos de selección y reemplazo de los individuos en las poblaciones.

5 Sistemas de Detección de Intrusos

5.1 Acerca de los Sistemas de Detección de Intrusos

Dentro de las soluciones tecnológicas para reforzar la seguridad de una red, los firewalls son muy populares. Un firewall es un sistema encargado del cumplimiento de las políticas de control de acceso a la red protegiéndola contra ataques que provengan del exterior. Sin embargo, debe considerarse que un atacante puede ocasionalmente descubrir alguna forma de burlar el control del firewall, o que el ataque puede provenir del interior de la organización.

Por lo tanto, la instalación de un firewall no es suficiente para mantener la red segura. Es necesario monitorear constantemente la actividad de la red en busca de la detección de posibles ataques o intrusiones lo más tempranamente posible para minimizar cualquier daño que puedan ocasionar en el sistema.

Una intrusión es cualquier conjunto de acciones que puede comprometer la integridad, confidencialidad o disponibilidad de una información o un recurso informático.

La detección de intrusos no se utiliza para reemplazar las técnicas de prevención tales como la autenticación y el control de accesos, sino en combinación con las medidas de seguridad existentes para evitar las acciones que intentan rebasar los sistemas de control de la seguridad. Por lo tanto, la detección de intrusos, se considera habitualmente como una segunda línea de defensa de las redes de computadoras.

La detección de intrusos consiste en un conjunto de métodos y técnicas para revelar actividad sospechosa sobre un recurso o conjunto de recursos computacionales. Es decir, eventos que sugieran comportamientos anómalos, incorrectos o inapropiados sobre un sistema [Lehmann]. El NIST (National Institute of Standards and Technology) define la detección de intrusos como el seguimiento de los eventos que suceden en un sistema informático con el objeto de hallar, a partir de su análisis, signos de intrusiones.

La detección de intrusos es una tarea compleja debido al continuo descubrimiento de vulnerabilidades explotadas por los atacantes y al aumento de la complejidad del hardware y software empleado. Por lo tanto es sumamente importante contar con herramientas informáticas que asistan al encargado de seguridad en esta tarea. Hoy día existen en el mercado una buena cantidad de productos conocidos como IDS -Intrusion Detection System-. Un IDS puede ser descrito como un detector que procesa la información proveniente del sistema monitoreado. Es una herramienta de apoyo en procesos de auditoría, entendida como el control del funcionamiento de un sistema a través del análisis de su comportamiento interno.

Para detectar intrusiones en un sistema, los IDS utilizan tres tipos de información: la recopilada tiempo atrás que tiene datos de ataques previos, la configuración actual del sistema y finalmente la descripción del estado actual en términos de comunicación y procesos [Debar].

La detección de intrusos se puede realizar a partir de la caracterización anómala del comportamiento y del uso que se hace de los recursos del sistema. Este tipo de detección pretende cuantificar el comportamiento normal de un usuario.

La idea central de este tipo de detección es el hecho de que la actividad proveniente de intrusos es un subconjunto de las actividades anómalas. Es decir que si un intruso consigue entrar al sistema seguramente su actividad no será como la de un usuario normal.

Los detectores de comportamiento anómalo presentan un alto costo computacional ya que requieren evaluar distintas métricas para determinar cuánto se aleja el comportamiento del usuario de lo que se considera normal.

Estos sistemas basan su funcionamiento en la recolección y análisis de información de diferentes fuentes, que luego utilizan para determinar la posible existencia de un ataque o penetración de intrusos.

En caso de que exista la suficiente certeza de la detección de un incidente, el IDS tiene como función principal alertar al administrador o personal de seguridad, para que tome acciones al respecto. Otras implementaciones más complejas son capaces de ir más allá de la notificación de un posible ataque, es decir pueden ejecutar acciones automáticas que impidan el desarrollo de la intrusión.

5.2 Soluciones con Redes Neuronales Artificiales

Si bien las computadoras actuales han demostrado ser capaces de resolver cálculos matemáticos complejos a enormes velocidades, muestran muchas dificultades en la realización de tareas sencillas para los seres humanos, como por ejemplo la identificación de un rostro conocido entre muchos otros.

El cerebro humano es capaz de interpretar información imprecisa suministrada por los sentidos a un ritmo extremadamente veloz. Su desempeño, altamente paralelo, no se ve afectado por la pérdida diaria de neuronas. Pero sin duda, su característica principal radica en su capacidad de aprender sin instrucciones explícitas de ninguna clase.

Basados en la eficiencia de los procesos llevados a cabo por el cerebro, e inspirados en su funcionamiento, varios investigadores han desarrollado la teoría de las Redes Neuronales Artificiales que actualmente constituyen la disciplina más representativa de la Inteligencia Artificial Subsimbólica. Éstas emulan a las biológicas,

y se han utilizado para aprender estrategias de solución basadas en ejemplos típicos de patrones.

A diferencia de los sistemas de cómputo tradicionales, las redes neuronales artificiales no ejecutan instrucciones sino que responden en paralelo a las entradas que se les presentan. El conocimiento de una red neuronal no se almacena en instrucciones, sino que se representa a través de su topología y de los valores de las conexiones -pesos- entre neuronas.

A diferencia de los sistemas basados en conocimiento -simbólico-deductivos-, las redes neuronales artificiales no basan su funcionamiento en la ejecución de reglas definidas a priori a través de un programa sino que son capaces de extraer sus propias reglas a partir de los ejemplos de aprendizaje. Esto es lo que les permite modificar su comportamiento en función de la experiencia.

Una vez que el conocimiento ha sido adquirido, se le pueden presentar a la red datos distintos a los usados durante el aprendizaje. La respuesta obtenida dependerá del parecido de los datos con los ejemplos de entrenamiento. Esto representa la importantísima capacidad de generalización.

Si bien se considera a las Redes Neuronales como el producto de una línea de investigación moderna, en realidad, son más antiguas que la propia Inteligencia Artificial Clásica -o Simbólica-. A la discusión sobre cuál debería ser la fecha considerada como el nacimiento de la Inteligencia Artificial, se agrega la opinión de quienes afirman que esta disciplina nace con el trabajo de Warren McCulloch y Walter Pitts que, a partir de observaciones biológicas, en 1943 presentaron el primer modelo de una neurona artificial, siete años antes del famoso artículo de Turing mencionado tempranamente en este trabajo.

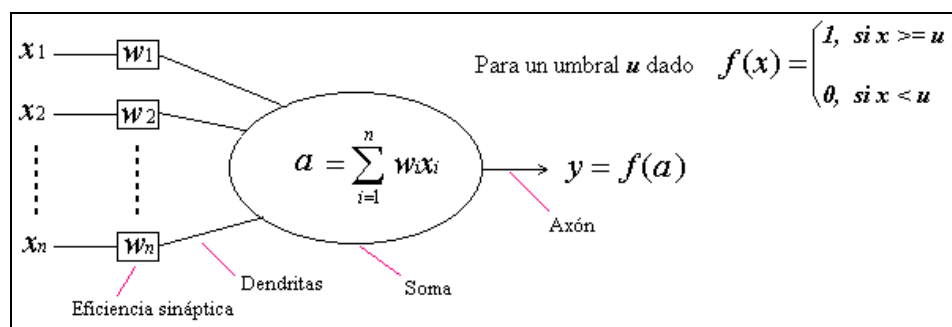


Figura 4 Modelo de neurona artificial de McCulloch y Pitts

Sin embargo, el primero en encontrar una aplicación práctica de la neurona de McCulloch y Pitts, y empleando las ideas de aprendizaje de Hebb⁶, fue el psicólogo Frank Rosenblatt quien diseñó la red tipo Perceptrón en el año 1957. El primer modelo de Perceptrón fue desarrollado imitando el funcionamiento del ojo humano y era básicamente un dispositivo de aprendizaje.

Cuando la investigación sobre Redes Neuronales parecía prometedora, en 1969 Minsky y Papert publicaron un trabajo que significó el abandono masivo de las prácticas de investigación en el área, sobre todo en los Estados Unidos, ya que en Europa su influencia fue menor. La crítica de estos investigadores estaba centrada en la limitación del Perceptrón -primera red neuronal con capacidad de aprendizaje- para aprender a resolver una operación lógica tan básica como el Or-exclusivo o XOR. Durante un período de 15 años, las instituciones no se atrevieron a financiar ningún trabajo que girara alrededor de las Redes Neuronales Artificiales, ya que parecía un camino sin salida.

Sin embargo, a principios de los 80, esta disciplina cobró un nuevo impulso a partir del redescubrimiento del exitoso algoritmo Backpropagation y junto con él se establecieron nuevas arquitecturas y estrategias de entrenamiento basadas en diversas ideas. En este nuevo período de resurgimiento de la neurocomputación, se ha encontrado en otra disciplina de la Inteligencia Artificial Subsimbólica ya mencionada en este trabajo, una de las mejores formas de conseguir redes neuronales con distintos grados de complejidad: los Algoritmos Evolutivos. Esta combinación de paradigmas suele denominarse Neuroevolución.

Exponer las particularidades de las distintas arquitecturas neuronales, su fundamentación matemática y sus algoritmos específicos de entrenamiento excede ampliamente los objetivos del presente trabajo. Para obtener una descripción detallada puede consultarse [Hecht-Nielsen 1991] y [Freeman 1993]

Existen numerosas publicaciones que exponen soluciones utilizando redes neuronales artificiales para la detección de intrusos y ello se debe a su flexibilidad

⁶ En pocas palabras, Hebb propuso que el aprendizaje sólo sería posible si existiese una simultaneidad temporal del funcionamiento de dos neuronas conectadas entre sí, una presináptica y otra postsináptica. Ese funcionamiento simultáneo, permite la conformación de circuitos neuronales y sería el factor *sine qua non* para la formación del recuerdo. Desafortunadamente Hebb carecía de las herramientas tecnológicas que le permitiesen poner a prueba sus ideas. Su teoría fue expuesta en el libro *Organization of Behaviour* en 1949.

y adaptación a los cambios que se pueden dar en el entorno y, sobre todo a la capacidad de detectar instancias de los ataques conocidos.

Lo que se puede cuestionar de los sistemas basados en redes neuronales es que éstas no constituyen un modelo descriptivo. Es decir, se puede observar cuál es la decisión tomada pero es muy difícil, conocer la razón que lleva a la red neuronal a tomarla⁷. Esta particularidad es especialmente inquietante en un sistema de detección de intrusos. Es posible que el administrador responsable de la seguridad de la red quiera conocer cuáles son los motivos por el cual una actividad es reportada por el sistema como una intrusión.

Uno de los primeros modelos de detección de intrusos basado en redes neuronales fue presentado en [Fox 1990] como método para crear perfiles de comportamiento de usuarios. Al igual que en [Debar 1992] se utilizaron redes neuronales que, basándose en una secuencia previa de comandos, son capaces de predecir la siguiente acción del usuario. El aprendizaje se realiza mediante redes neuronales recurrentes que están continuamente observando la actividad y que son capaces de olvidar comportamientos antiguos. La topología de las redes neuronales recurrentes se caracteriza por la presencia de conexiones que generan ciclos. Éstos hacen posible cierta capacidad de memoria de corto plazo.

Otro trabajo presentado más tarde [Cansian 1997] utiliza una red neuronal para identificar el comportamiento proveniente de intrusos a partir de la observación de patrones capturados de una red.

En [Hoglund 1998] se propone el uso de dos mapas auto-organizativos (SOM)⁸. Mientras que el primero se utiliza como método de visualización del comportamiento de los usuarios, el otro compara el comportamiento de un usuario con el perfil generado en la red anterior ayudado con medidas estadísticas.

En [Grediaga 2002] se presenta un IDS basado en redes neuronales utilizando dos arquitecturas distintas: una red *backpropagation* y una red SOM. Con su

⁷ Existe una línea de investigación en el área de Data Mining que estudia, para ciertas arquitecturas neuronales específicas, la manera de deducir las reglas que gobiernan la actividad de la red neuronal, en forma entendible para un ser humano.

⁸ La red neuronal SOM (Self Organizing Maps) fue propuesta por Kohonen y por su naturaleza competitiva, ha sido utilizada para resolver una amplia gama de problemas de clustering y clasificación. Puede encontrarse una descripción detallada de esta arquitectura y otras competitivas en [Kohonen 1997]

propuesta los autores intentan paliar la principal limitación del IDS tradicional: su incapacidad para reconocer ataques ligeramente modificados respecto a los patrones con los que realiza las comparaciones. La idea aquí es explotar la capacidad de generalización de las redes neuronales, basada en el reconocimiento de los detalles significativos de los patrones aprendidos, para clasificar correctamente nuevos patrones nunca antes analizados. Según los resultados publicados se obtuvo un porcentaje de aciertos superior al 90% utilizando la red backpropagation y aún más alto con la utilización de la red SOM.

En la Universidad de Ohio se desarrolló un IDS de red llamado INBOUNDS (Integrated Network-Based Ohio University Network Detective Service) inicialmente basado en el análisis estadístico para la detección de anomalías y posteriormente reemplazado por un módulo de detección que utiliza mapas auto-organizativos.

Desde sus inicios, las redes neuronales artificiales han sido utilizadas exitosamente como dispositivos reconocedores de patrones. No sorprende entonces su buena adecuación al problema de detección de actividad generada por intrusos, la que puede caracterizarse por ciertos patrones de comportamiento. Una red neuronal puede aprender a reconocer estos patrones, aún si se presentan ligeramente camuflados.

Tanto la red neuronal backpropagation como los mapas auto-organizativos de Kohonen (red SOM) presentan ventajas que han sido bien aprovechadas por los IDSs que las utilizan.

La investigación en Redes Neuronales Artificiales constituye una actividad muy prolífera con aplicación en diversas áreas informáticas. En particular, han mostrado un buen desempeño en sistemas de detección de intrusos. Dada la importancia que actualmente tienen los aspectos de seguridad para los sistemas de información, es muy probable que sigan apareciendo implementaciones de IDSs que exploten las facilidades que ofrece esta importante disciplina de la Inteligencia Artificial.

Comentaré finalmente que las redes neuronales artificiales también han sido utilizadas con éxito en otras cuestiones inherentes a las redes de datos como ser el control en sistemas de comunicación [Morris 1994], el control de congestión en redes

ATM [Tarraf 1995], y la recuperación de información basada en contenido en redes P2P [Maxim 2003].

5.3 Soluciones con Sistemas Inmunes Artificiales

Los Sistemas Inmunes Artificiales están inspirados en el sistema inmune natural. Este es el encargado de la defensa del organismo; controla lo que se introduce en él y lo identifica como propio o no propio. Son varias las características del sistema inmune que resultan de interés para muchos modelos computacionales: capacidad de reconocer cuerpos extraños y anomalías, detección distribuida y tolerancia al ruido -no es necesario una detección exacta de los cuerpos extraños-, y además es capaz de memorizar y reforzar conocimiento sobre los organismos infecciosos -agentes patógenos- aprendidos.

El sistema inmune está formado por moléculas, células y órganos que están en continua vigilancia de los organismos infecciosos. Cuando un nuevo agente patógeno entra al organismo, el sistema inmune lo detecta, inmoviliza y destruye, pero además lo "memoriza" para poder neutralizarlo más eficientemente la próxima vez que deba enfrentarse con él.

Hay dos formas en las que el cuerpo identifica material extraño: el sistema inmune innato y el sistema inmune adquirido. La respuesta inmune innata está basada en la *selección negativa*, y la adquirida en la *selección clonal*. Ambas selecciones son importantes en los modelos artificiales de sistemas inmunes.

La selección negativa interviene en la obtención de las células llamadas linfocitos T que son las responsables de identificar lo *propio* de lo *ajeno*. Estas células se desarrollan mediante un proceso de variación genética pseudo-aleatoria y luego son sometidas a un proceso de selección negativa donde son destruidas todas aquellas que reaccionan contra sustancias propias del organismo. Obsérvese que este proceso no necesita información alguna sobre ningún agente patógeno.

En el contexto de los Sistemas Inmunes Artificiales, y más específicamente en los IDS, los modelos de selección negativa, consisten básicamente en distinguir lo propio -usuarios legítimos, comportamiento normal, archivos no corruptos, etc.- del resto -usuarios no autorizados, virus anomalías, etc.-.

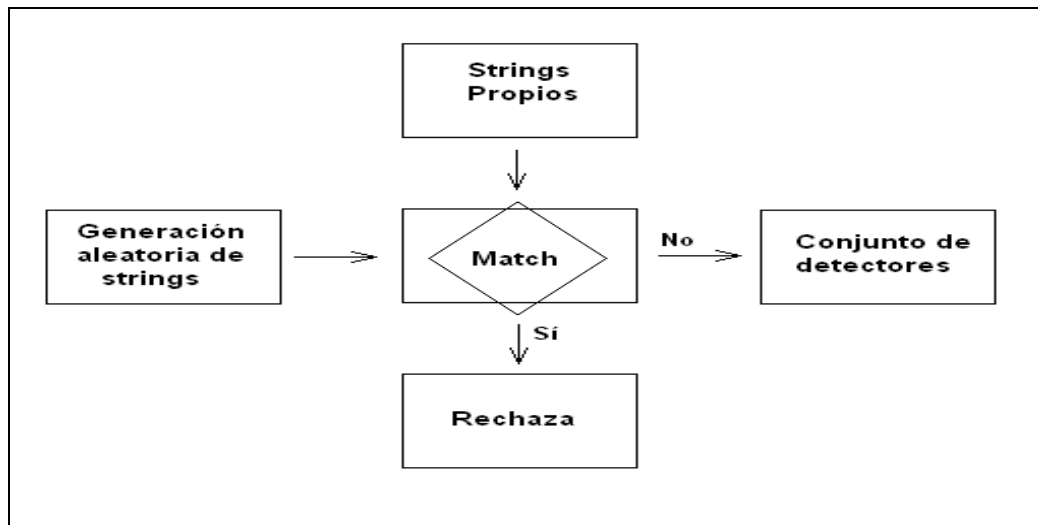


Figura 5 Generación de detectores por Selección Negativa

El principio de la selección clonal plantea el mecanismo que explica cómo se generan los anticuerpos específicos, representantes de la inmunidad adquirida, contra una gran variedad de antígenos -nombre que se dio a cualquier sustancia que induzca la formación de anticuerpos- que pone en funcionamiento el sistema inmune ante el estímulo de un antígeno específico. Aunque existen muchas células capaces de producir anticuerpos diferentes -los linfocitos B-, este principio establece que aquellos linfocitos B con mayor afinidad con el invasor sufren un proceso de clonación con alta tasa de mutación. Así, estos linfocitos secretan gran cantidad de anticuerpos que eliminan y neutralizan a los invasores. Parte de esta población de linfocitos permanece en el organismo por un largo período como células de memoria, de manera que, cuando se presenta el mismo tipo de antígeno o uno similar, la respuesta del sistema inmune será más rápida y eficiente. A esta respuesta se le llama *respuesta secundaria*.

Los principales aspectos de la selección clonal que son tomados por los sistemas inmunes artificiales son: mantenimiento de un conjunto específico de memoria, clonación y selección de los más aptos, maduración de afinidad y reelección de genes proporcional a su afinidad antigénica, generación y mantenimiento de la diversidad.

La idea de utilizar principios inmunológicos en la seguridad informática se inició en 1994. Forrest y su grupo de la universidad de Nuevo México trabajaron desde entonces en la creación de un sistema inmune artificial [Forrest 1994]. En 1996 este grupo realizó su primer experimento de detección de intrusiones a partir de llamadas al sistema de procesos UNIX [Forrest 1996]. El sistema recoge información de

secuencias de comandos del agente de correo sendmail de UNIX, y la utiliza en el período de entrenamiento para definir lo que es propio.

En [Hofmeyr 2000] se presenta un IDS llamado LISYS basado en un sistema inmune artificial, que resulta efectivo para detectar intrusiones manteniendo un nivel bajo de falsos positivos. En [Balthrop 2002] se analiza el desempeño de LISYS considerando los parámetros más importantes que impactan sobre la capacidad de detección y la cantidad de falsos positivos.

Otro grupo de investigadores desarrollaron ADENOIDS un IDS híbrido que, según sus autores, posee las mismas capacidades de aprendizaje y adaptación que el sistema inmune humano. Su trabajo está descrito en [Paula 2002] y en [Paula 2004]

Existen muchas publicaciones con estrategias inspiradas en el sistema inmune para implementar IDSs. La mayoría de ellas se concentran en la construcción de modelos y algoritmos para la detección de intrusiones basados en el reconocimiento de patrones de comportamientos.

Los sistemas inmunes artificiales han demostrado ser útiles y adecuarse bien a las necesidades en el área de seguridad informática. Constituyen una de las técnicas más recientes incorporadas a la construcción de sistemas de detección de intrusos y seguramente continuarán siendo motivo de estudio e investigación.

6 Recuperación de la Información en Internet

6.1 *Acerca de la Recuperación de la Información en Internet*

En los últimos años, los métodos para gestionar información han sufrido grandes transformaciones debido a la aparición de Internet, y sobre todo, de la World Wide Web. A medida que éstas han ido creciendo y evolucionando se ha complicado la forma de localizar y consultar información existente en la propia red. La mayoría de las herramientas disponibles actualmente para la recuperación de información se centran más en la cantidad que en la calidad de la propia información. Es habitual que el usuario se sienta desbordado por el elevado volumen de documentos o de referencias que se suministran. Existe entonces la necesidad de una cierta automatización que facilite la labor de localización de información de calidad y que a su vez permita diferenciar entre

ella desde el punto de vista subjetivo del propio usuario. En este punto, el empleo de técnicas inteligentes resulta adecuado para la mejora de los resultados obtenidos y es justo en este momento donde cabe plantearse el uso de agentes inteligentes [Mas 2005].

6.2 Soluciones con Agentes Inteligentes

Un agente es un sistema informático, situado en un entorno, capaz de realizar acciones de forma autónoma para conseguir sus objetivos de diseño [Wooldridge, 2002]. Los agentes de búsqueda que encuentran automáticamente información en la red según las especificaciones del usuario, se denominan también agentes inteligentes, agentes personales, o knowbots. En general para que un agente sea considerado inteligente debe exhibir alguna forma de razonamiento y/o aprendizaje.

Los agentes inteligentes de información o agentes de Internet pueden definirse como aquellos sistemas de computación que tienen acceso a múltiples y heterogéneas fuentes de información que están distribuidas geográficamente [Klusch 2001].

Un ejemplo clásico de agente de información es *Letizia* [Liebarman] desarrollado en el MIT Media Laboratory. Es un agente de filtrado con un funcionamiento básicamente proactivo. Trabaja con un navegador convencional utilizando una configuración multi-ventana. Mientras que un usuario está consultando una página Web, *Letizia* realiza una búsqueda en profundidad desde la página actual, visualizando los enlaces encontrados. Así va proporcionando de forma continua recomendaciones sobre páginas web.

Para que su trabajo sea eficiente, el agente debe conocer al usuario para anticiparse a sus opiniones sobre una página determinada. Para ello, *Letizia* observa el comportamiento del usuario, e intenta inferir sus gustos automáticamente a partir del almacenamiento de cada una de las opciones tomadas y de cada página que éste haya leído.

También desarrollado dentro del MIT Media Laboratory encontramos *ExpertFinder* [Vivacqua 2000], un sistema multiagente que ayuda a un usuario a encontrar expertos que puedan asistirlo en un problema concreto. Es un sistema de recomendación dónde cada usuario tiene un agente que conoce sus áreas y niveles de

experiencia junto con la de los demás usuarios. Los agentes construyen automáticamente el perfil de cada usuario e intercambian esta información. En caso de necesitarlo, el agente debe determinar quién es el mejor experto disponible para ayudar en una situación particular. En definitiva ExpertFinder intenta construir comunidades de expertos que intercambian información aumentando el nivel de conocimiento de la comunidad de forma más rápida y eficiente que mediante transacciones uno a uno.

Amalthea [Moukas 1998] es otro sistema desarrollado en el MIT Media Laboratory. Básicamente consiste en una población de agentes que evoluciona para la búsqueda y filtrado de información. Cada agente actúa como un filtro muy especializado que cubre un área de interés. El sistema intenta presentar al usuario páginas y noticias de su predilección. Tras una consulta el usuario explora los resultados de búsqueda generados y proporciona realimentación al sistema mediante la puntuación de la relevancia de cada ítem. Con esa información, *Amalthea* aprende, por medio de la evolución de la población de agentes, las necesidades del usuario y se adapta según van cambiando los intereses del mismo.

Los *infoSpiders* [Menczer 2000] son agentes adaptativos que exploran documentos decidiendo de forma autónoma qué enlaces seguir, ajustando su estrategia al contexto y a las preferencias del usuario. El entorno de información por el que se mueve el agente es la Web y los discos locales. El usuario proporciona una lista de términos y de sitios iniciales -obtenidas por cualquier motor de búsqueda tradicional-. En cada sitio de la lista se coloca un agente que, estimando la relevancia de los enlaces, comienza la navegación. Cada agente posee cierta cantidad de energía que va consumiendo al desplazarse. Si los documentos visitados por el agente son relevantes, éste es recompensado con más energía. La salida del sistema es una lista de enlaces a documentos ordenados por su relevancia estimada.

Existe una gran cantidad de sistemas basados en agentes para la recuperación de la información en Internet que podría citar en este punto. Sin embargo creo conveniente evitar las largas exposiciones y no sobreabundar en ejemplificaciones. En cambio comentaré que el lector interesado puede hallar varios artículos relacionados con este tema en el workshop "Multi-Agent Information Retrieval and Recommender Systems" perteneciente a la "Nineteen International Joint Conference on Artificial

Intelligence" realizado en 2005, estando todos los artículos accesibles para descargar desde la url http://www.iro.umontreal.ca/~aimeur/IJCAI-2005/IJCAI_workshop.htm.

7 Conclusiones

La aplicación de conceptos pertenecientes al área de la Inteligencia Artificial ha dado origen a los sistemas denominados inteligentes. Ellos son capaces de encontrar soluciones convenientes, en tiempos aceptables, a problemas no decidibles o pertenecientes al conjunto NP. Encontramos sistemas inteligentes prácticamente en cualquier área de la informática, y por supuesto, los problemas inherentes a las redes de computadoras no constituyen la excepción.

Entre las metaheurísticas más utilizadas se encuentran los Algoritmos Evolutivos. Existe sobrada evidencia empírica sobre su capacidad para realizar una buena exploración del espacio de búsqueda. Así, cuando se usan para problemas de optimización, se ven menos afectados por los máximos locales que las técnicas tradicionales. En consecuencia, no resulta extraño que se hayan utilizado exitosamente para optimizar el diseño, el ruteo y la transmisión multicast en redes de datos. El aprendizaje evolutivo ha sido empleado también en la construcción de agentes inteligentes que asisten al usuario en la recuperación de información en la Internet.

Los Algoritmos Evolutivos constituyen un área de investigación muy activa con aplicación práctica en muchas cuestiones relacionadas con las redes de datos. Debido al gran interés que estas últimas despiertan en el mundo actual, es indudable que seguirán apareciendo nuevas soluciones basadas en evolución que mejoren algún aspecto de las redes de comunicaciones.

Los sistemas ACO –Ant Colony Optimization- basados en el comportamiento social de las hormigas también han sido utilizados con buenos resultados en el área de las redes de datos. La mayor cantidad de propuestas corresponden a soluciones al problema de ruteo de la información pero también se han utilizado en otras cuestiones como la búsqueda distribuida de fallas, la localización de servicios en redes Ad-Hoc y la búsqueda de recursos en redes P2P.

Las redes neuronales artificiales han demostrado un alto rendimiento como dispositivos reconocedores de patrones. Además aprenden por medio de ejemplos,

por lo que no es necesario programarlas. Una vez completado el aprendizaje, son capaces de clasificar patrones correctamente aún en presencia de ruido o con información incompleta. Estas características han llamado la atención de los diseñadores de Sistemas de Detección de Intrusos quienes comenzaron a utilizarlas a partir de los años '90. Después de todo, la actividad que generan los intrusos puede entenderse como un patrón de comportamiento que las redes neuronales son capaces de aprender a identificar.

Las arquitecturas neuronales más utilizadas en esta área han sido la backpropagation y los mapas auto-organizativos de Kohonen cuyas capacidades de reconocimiento han sido bien aprovechadas por los IDSs que las utilizan. Debido a la importancia que siempre tienen los aspectos de seguridad en los sistemas informáticos, es muy probable que sigan apareciendo implementaciones de IDSs que exploten las ventajas ofrecidas por las redes neuronales.

Las redes neuronales artificiales también han sido utilizadas en otras áreas como ser el control de congestión en redes ATM, el control en sistemas de comunicación y la recuperación de información basada en contenido en redes P2P.

Los Sistemas Inmunes Artificiales constituyen una de las técnicas bio-inspiradas de aparición más reciente. A mediados de los '90 comenzaron a aplicarse en seguridad informática para la construcción de sistemas de detección de intrusos. En ellos se han utilizado los modelos de selección negativa para distinguir lo propio -usuarios legítimos, comportamiento normal, archivos no corruptos, etc.- del resto -usuarios no autorizados, virus anomalías, etc.-.

Los Sistemas Inmunes Artificiales han demostrado ser útiles y capaces de adecuarse bien a las necesidades en el área de seguridad informática y seguramente continuarán arrojando buenos resultados.

Por último, los sistemas basados en agentes han mostrado ser una opción prometedora como asistentes para la búsqueda y recuperación de información en la Internet. Ellos automatizan el proceso que facilita la localización de información de calidad desde la apreciación subjetiva del propio usuario. Muchos son capaces de inferir automáticamente el perfil del usuario y adaptarse por aprendizaje cuando se advierte algún cambio de actitud o de interés por parte de éste. Constituyen así una herramienta

sumamente útil prestando un valioso servicio para la recuperación inteligente de información.

De esta manera finalizo la presente monografía. Considero que cumple con el objetivo planteado presentando un panorama general sobre la aplicación de sistemas inteligentes en el área de las redes de datos. Espero además que las referencias a trabajos de investigación citados en cada uno de los temas abordados, sirvan para que el lector interesado pueda profundizar en los detalles que estime conveniente consultando las fuentes originales.

8 REFERENCIAS

- [Amin 2003] Amin Kaizar A. Y Mikler Armin R. Dynamic Agent Population in Agent-Based Distance Vector Routing. *Neural, Parallel & Scientific Computations*. Pp 127-142 (2003)
- [Ardenghi 2005] Ardenghi, Jorge. Planificación de Procesos en Sistemas Distribuidos. Módulo 4 del curso de Sistemas Distribuidos del magister en Redes de Datos de la facultad de Informática de la UNLP.
- [Balthrop 2002] Balthrop Justin, Forrest Stephanie and Glickman Matthew. Revisiting LISYS: Parameters and Normal Behavior. *Evolutionary Computation*, 2002. CEC '02. Vol 2, pp: 1045-1050. Honolulu, HI, USA. (2002)
- [Barán 2001] Barán, Benjamín y Sosa, Rubén. AntNet Routing Algorithm for Data Networks based on Mobile Agents. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. Nro. 12 pp. 75-84 ISSN: 1137-3601 (2001).
- [Cansian 1997] Cansian, A.M., S. Moreira, E., Carvalho, A. and Bonifacio Jr, J. M. Network Intrusion Detection using Neural Networks. *International Conference on Computational Intelligence and Multimedia Applications (ICCMA'97)* 1997
- [Colouris 2001] Colouris, G. F.; Dollimore, J. y T. Kindberg; "Distributed Systems: Concepts and Design". 3rd Edition. Addison-Wesley, 2001.
- [Comer 1995] Comer Douglas E. *Redes Globales de Información con Internet y TCP/IP Principios básicos, protocolos y arquitectura*. Prentice-Hall Hispanoamericana 1995.
- [CottaPorrás 1998] Cotta Porrás, Carlos. Tesis doctoral "Un Estudio de las Técnicas de Hibridación y su Aplicación al Diseño de Algoritmos Evolutivos". Departamento de Lenguajes y Cincias de la Computación. Universidad de Málaga. 28 de Mayo de 1998.
- [Dan 2004] Dan, Wang. A Resource Discovery Model based on Multi-Agent Technology in P2P System. *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT '04)*

- [de Araujo 2002] de Araujo, P., Barbosa G. Multicast Routing With Auality of Service and Traffic Engineering Requirements in the Internet, Based On Genetic Algorithm. Proceedings of the VII Brazilian Symposium on Neural Networks (SBRN'02), 2002.
- [Debar 1992] Debar, H., Becker, M., Siboni, D. A Neural Network Component for an Intrusion Detection System. IEEE Symposium on Research in Computer Security and Privacy, 1992
- [Debar] Debar, H. "An introduction to Intrusion-Detection Systems". IBM Research, Zurich Research Laboratory.
- [Dengiz 1997a] Dengiz Berna, Fulya Altipamak. "Efficient optimization of all-terminal reliable network using evolutionary approach". IEEE Transation on Reliability, march 1997.
- [Dengiz 1997b] Dengiz, Berna. "Efficient Optimization of All Reliable networks, using an Evolutionary Approach". (IEEE transaction on reliability vol 46 nro 1 1997).
- [Di Caro 1998] Di Caro, Gianni; Dorigo, Marco. Two ant colony algorithms for best-effort routing in datagram networks. In Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98), pp.541-546 IASTED/ACTA Press.
- [Dorigo 1991] Dorigo, M. Maniezzo, V., Colorni, A. The Ant System: An Autocatalytic Optimizing Process. Tech. Rep. 91-016, Dipartimento di Elettronica ed Informatca. Politecnico di Milano (IT), 1991.
- [Dorigo 1992] Dorigo, Marco. Optimization, Learning and Natural Algorithms. PhD thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano IT, 1992.
- [Dorigo 1997] Dorigo, M. Di Caro, G. AntNet. A Mobile Agents Approach to Adaptive Routing. Technical Report. IRIDIA Free Brussels University. Belgium (1997)
- [Dorigo 1999] Dorigo Marco; Di Caro, Gianni; Gambardella, Luca M. Ant Algorithms for Discrete Optimization. Artificial Life 5, 137-172, 1999 Massachusetts Institute of Technology.
- [Forrest 1994] Forrest, S. Perelson, A. Allen L. And Cherukuri R. Self-nonsel self discrimination in a computer. In proceedings of the IEEE Symposium on Research in Securityand Privacy, pp. 202-212, (1994).
- [Forrest 1996] Forrest, S., Hofmeyr, S., Somayaji, A., Longstaff T., A Sense of Self for UNIX Processes. IEEE Symposium on Computer Security and Privacy, 1996
- [Fox 1990] Fox, K., Henning, R., Redd, J., Simonian, R. A. Neural Network Approach Towards Intrusion Detection. 13th National Computer Security Conference, 1990.
- [Freeman 1993] Freeman, James A. & Skapura, David M. "Redes neuronales Algoritmos, aplicaciones y técnicas de programación". Addison-Wesley, 1991. Versión en español de: Rafael García -Bermejo Giner. Addison-Wesley Iberoamericana 1993.
- [Garey 2000] Garey Michael R., Johnson David S. "Computers and Intractability A Guide to de Theory of NP-Completeness" W.H. Freeman and Company. 2000.
- [Goldberg 1989] Goldberg, David.E. "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.

- [Grediaga 2002] Grediaga, Angel. Ibarra, Francisco. Ledesma, Bernardo. Brotons, Francisco. Utilización de redes neuronales para la detección de intrusos. Primer Congreso Iberoamericano de Seguridad Informática CIBSI '02. Morelia, México. febrero de 2002
- [Hamdam 2004] Hamdam, M., El-Hawary, M. E. Multicast Routing with Delay and Delay Variation Constraints Using Genetic Algorithm. CCECE 2004- CCGEI 2004, Niagara Falls, May/mai, pp. 2263-2266. (2004)
- [Hecht-Nielsen 1991] Hecht-Nielsen, Robert. "Neurocomputing". Addison-Wesley Publishing Company. 1991
- [Hofmeyr 2000] Hofmeyr, S., Forrest, S. Architecture for an Artificial Immune System. Evolutionary Computation, 2000.
- [Hoglund 1998] Hoglund, A., Hatonen, K., AND Tuononen, T. A UNIX Anomaly Detection System Using Self-ORGANISING Maps. Recent Advances in ITRUSION Detection, RAID 1998.
- [Hopcroft 1993] Hopcroft John E., Ullman Jeffrey D. "Introducción to Automata Theory Languages, and Computation" 1993.
- [Jha 2005] Jha, Ruchir. A Routing Algorithm Inspired from a Distributed Autonomous Multi-Agent System -The Ant Colony. Proceedings 19th European Conference on Modelling and Simulation Yuri Merkurjev, Ricard Zobel, Eugene Kerckhoffs. ECMS, 2005 ISBN 1-84233-112-4 (Set) / ISBN 1-84233-113-2 (CD).
- [Karabi 2004] Karabi, Mehdi. Fathy, Mahamood. Dehghan, Mebdi. QoS Multicast Routing Based on a Heuristic Genetic Algorithm. CCECE 2004- CCGEI 2004, Niagara Falls, May/mai, pp. 1727-1730. (2004)
- [Klusch 2001] Klusch M. Information agent technology for the Internet: a survey. Data and Knowledge Engineering, Vol 36(3,) pp 337-372. (2001)
- [Kohonen 1997] Kohonen Teuvo. Self-Organizing Maps. ISBN 3-540-62017-6 2nd Edition Springer-Verlag. 1997
- [Konak 2005] Konak Abdullah, Smith Alice. "Designing resilient networks using a hybrid genetic algorithm approach" Proceedings of the 2005 conference on Genetic and evolutionary computation, June 25-29, 2005 (GECCO'05), Washington DC, USA
- [Lehmann] Lehmann, D. "Intrusion Detection FAQ". Disponible en: http://www.sans.org/resources/idfaq/what_is_id.php
- [Lieberman] Lieberman, Henry. Letizia: An Agent That Assists Web Browsing. Letizia Home Page url: <http://web.media.mit.edu/~lieber/Lieberary/Letizia/Letizia-Intro.html>
- [Magnago 2006] Magnago, Héctor Enrique. ALGORITMOS EVOLUTIVOS APLICADOS A PROBLEMAS DE DISEÑO DE REDES CONFIABLES. Tesis de Magister en Redes de Datos. Facultad de Informática. UNLP. Febrero de 2006.
- [Mas 2005] Mas, Ana. Agentes Software y Sistemas Multi-Agentes: Conceptos, Arquitecturas y Aplicaciones. Pearson Prentice Hall. 2005.
- [Maxim 2003] Maxim, Rodionov and Hui, Siu Cheung. Intelligent Content-Based Retrieval for P2P Networks. Proceedings of the 2003 International Conference on Cyberworlds (CW' 03) 2003 IEEE.

- [Morris 1994] Morris, Robert J. T.; Samadi, Behrokh. Neural Network Control of Communications Systems. IEEE Transactions on Neural Networks, Vol. 5 nro. 4 (1994).
- [Moukas 1998] Moukas A., Maes P. Amalthea: An Evolving Multiagent Information Filtering and Discovery System for de WWW. Invited paper for the first issue of the Journal of Autonomous Agents and Multi-Agent Systems.
- [Munetomo 1998] Munetomo, Masaharu; Takai, Yoshiaki and Sato, Yoshiharu. A Migration Scheme for the Genetic Adaptive Routing Algorithm. Proceedings of the 1998 IEEE Conference on Systems, Man, and Cybernetics, pages 219-227. 1998.
- [Munetomo 1999] Munetomo, Masaharu "Designing genetic algorithms for adaptive routing algorithms in the internet", booktitle: "Evolutionary Telecommunications: Past, Present, and Future" 1999.
- [Mykkeltveit 2004] Mykkeltveit A., Heegaard P.E., Wittner O. Realization of a Distributed Route Management System on Software Routers. Norsk Informatikkonferanse, Stavanger, Norway 2004. pp. 102-112. Tapir Akademisk Forlag. ISBN 82-519-2004-3 (2004)
- [Paula 2002] Paula, F.S.; Reis, M.A.; Fernandes, D.A.M.; Geus, P.L.; Adenoids: a hybrid IDS based on the immune system. ICONIP '02. Proceedings of the 9th International Conference on Neural Information Processing. Computational Intelligence for the E-Age (IEEE Cat. No.02EX575) vol.3. pp.1479-1484 (2002)
- [Paula 2004] de Paula, F.S.; de Castro, L.N.; de Geus, P.L.; An intrusion detection system using ideas from the immune system. Congress on Evolutionary Computation, CEC2004 Portland, Oregon. Vol.1 pp.:1059-1066. (2004)
- [Ravikumar 1998] Ravikumar C., Bajpai R. Source-based delay bounded multicasting in multimedia networks. Computer Communications, Vol. 21, 1998, pág. 126-132.
- [Reichelt 2004] Reichelt, D.; Rothlauf, F.; Gmilkowsky P. "Designing Reliable Communication Networks with a Genetic Algorithm Using a Repair Heuristic" Lecture Notes in Computer Science. Book: Evolutionary Computation in Combinatorial Optimization. pp 177-187 (2004)
- [Tanenbaum 1995] Tanenbaum, Andrew S.; Distributed Operating Systems. Prentice Hall, 1995.
- [Tarraf 1995] Tarraf, A.A.; Habib, I.W.; Saadawi, T.N. Congestion control mechanism for ATM networks using neural networks. Conference on Communications, 1995. ICC 95 Seattle, Gateway to Globalization, 1995 IEEE International Volume 1, 18-22 June 1995 pp. 206 - 210
- [Vivacqua 2000] Vivacqua A.; Lieberman, H. Agents to Assist in Finding Help. In proceedings of the ACM Conference on Human Factors in Computing Systems, CHI 2000, 65-72. ACM Press.
- [Wooldridge 2002] Wooldridge, M. An Introduction to Multiagent Systems. John Wiley & Sons, LTD. 2002.