

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY



**Simulación Computacional del Efecto Baldwin
en un Modelo Depredador – Presa Artificial.**

TESIS QUE PRESENTA

ULISES BOLAÑOS LEPÉRE

**MAESTRÍA EN CIENCIAS COMPUTACIONALES
MCCI 98**

DICIEMBRE 5, 2003



193227

**Simulación Computacional del Efecto Baldwin
en un Modelo Depredador – Presa Artificial.**

TESIS QUE PARA OPTAR EL GRADO DE
MAESTRO EN CIENCIAS COMPUTACIONALES
PRESENTA

Ulises Bolaños Lepére

Asesor: Dr. Edgar Emmanuel Vallejo Clemente
Asesor Externo:

Comité de tesis: Dr. Maria de los Ángeles Junco Rey
Dr. Bedrich Benes

Jurado:	Dr. Maria de los Ángeles Junco Rey	Presidente
	Dr. Bedrich Benes	vocal
	Dr. Edgar Emmanuel Vallejo Clemente	vocal

Atizapán de Zaragoza, Edo. Méx., 5 de Diciembre de 2003.

RESUMEN

Las teorías de Lamarck, Darwin, y el efecto Baldwin involucran conceptos naturales como la evolución y el aprendizaje. La computación, por medio de técnicas computacionales ha tratado de simular el comportamiento natural para entenderlo y también para usarlo como inspiración para automatizar procesos que involucran clasificación y optimización. Técnicas como algoritmos genéticos y redes neuronales son herramientas de la computación que nos pueden ayudar a simular dicho comportamiento. En este trabajo utilizamos los algoritmos genéticos para optimizar el movimiento que provocan los pesos de la red neuronal de las presas en un simulador del efecto Baldwin y poder comprobar así la hipótesis de que el aprendizaje guía la evolución.

La teoría de Lamarck establece que el aprendizaje se codifica en el material genético, la teoría de Darwin surge en oposición a la teoría de Lamarck y afirma que el aprendizaje no se codifica en el material genético. En cambio, la teoría del efecto Baldwin relaja esta dicotomía ya que nos dice que el aprendizaje guía la evolución por medio del proceso de asimilación genética. Han existido algoritmos computacionales basados en estas tres teorías. Nuestro objetivo es comprobar la hipótesis de que el postulado del efecto Baldwin es válido y que los algoritmos evolutivos en los pesos de una red neuronal son una alternativa viable de optimización y clasificación para un sistema depredador-presa artificial.

En el simulador depredador-presa, las presas y el depredador habitarán en una malla bidimensional y toroidal. Las presas y el depredador estarán representados por triángulos para poder observar mejor su movimiento y dirección. Internamente, cada presa está representada por una red neuronal de cinco capas de entrada, dos capas intermedias y tres capas de salida. Esto hace que la red neuronal tenga 16 conexiones o pesos que van a ponderar las entradas o percepción para poder realizar salidas que indican el movimiento a realizar.

Se generó un algoritmo genético que va a evolucionar los pesos de dichas presas con una función de aptitud que depende del medio ambiente y se realizó un algoritmo de aprendizaje que modifica los pesos resultado de la evolución. Después se observaron los resultados e implicaciones de ambos algoritmos.

Efectivamente, los resultados experimentales confirman la validez de la hipótesis de que el aprendizaje guía la evolución del comportamiento de escape en un ambiente depredador-presa. También se comprueba que los algoritmos evolutivos con aprendizaje son mejores que los solamente evolutivos en cuanto a optimización para un sistema depredador-presa se refiere.

En este trabajo solamente nos enfocamos a las presas, por lo que sería interesante evaluar los procesos de coevolución también evolucionando los depredadores y observar los resultados.

ÍNDICE DE CONTENIDO

	Página
DEDICACIONES.....	ii
RECONOCIMIENTOS.....	iii
RESUMEN.....	iv
ÍNDICE DE TABLAS.....	vii
ÍNDICE DE ILUSTRACIONES O FIGURAS.....	viii
 Capítulo	
1. INTRODUCCIÓN.....	11
1.1 Generalidades.....	11
1.2 Problema planteado.....	13
1.3 Objetivos de investigación e hipótesis.....	14
1.4 Contribuciones esperadas.....	15
1.5 Organización del documento.....	15
2. TRABAJOS RELACIONADOS.....	16
2.1 Vida artificial.....	16
2.2 Biología y evolución.....	17
2.2.1 Formas de adaptación de los sistemas biológicos.....	17
2.3 Computación.....	18
2.3.1 Algoritmos evolutivos.....	18
2.3.1.1 Algoritmo genético simple.....	18
2.3.2 Redes neuronales.....	20
2.3.2.1 Fisiología de una neurona simple.....	21
2.3.2.2 Conexiones sinápticas.....	23
2.3.2.3 Redes neuronales artificiales.....	23
2.3.2.4 Retropropagación(“back propagation”).....	26
2.3.3 Evolución de redes neuronales artificiales.....	30
2.3.3.1 Evolución de pesos de conexión.....	30
2.4 Trabajos previos.....	30
3. TRABAJO PROPUESTO.....	35
3.1 Ambiente.....	35
3.1.1 Presas.....	36
3.1.2 Depredador.....	38
3.2 Representación interna.....	40
3.2.1 Presas.....	40
3.2.1.1 Especificación de los pesos de conexión.....	42
3.2.1.2 Codificación decimal de los pesos.....	43
3.2.2 Depredador.....	44
3.3 Fases del trabajo propuesto.....	45

3.3.1 Evolución.....	46
3.3.1.1 Proceso de evolución.....	46
3.3.1.1.1 Inicializar los cromosomas aleatoriamente.....	46
3.3.1.1.2 Selección de los cromosomas más aptos según su función de aptitud.....	46
3.3.1.1.3 Cruzamiento.....	47
3.3.1.1.4 Mutación.....	48
3.3.1.1.5 Guardar los pesos resultado de la generación.....	48
3.3.2 Evolución con aprendizaje(efecto Baldwin).....	49
3.3.2.1 Proceso de evolución con aprendizaje.....	49
3.3.2.1.1 Inicializar los cromosomas aleatoriamente.....	49
3.3.2.1.2 Selección de los cromosomas más aptos según su función de aptitud.....	49
3.3.2.1.3 Cruzamiento.....	50
3.3.2.1.4 Mutación.....	51
3.3.2.1.5 Guardar los pesos resultado de la generación.....	51
3.3.3 Simulación.....	51
4. EXPERIMENTOS Y RESULTADOS.....	52
4.1 Evolución.....	52
4.1.1 Experimentos y resultados.....	52
4.2 Evolución con aprendizaje(efecto Baldwin).....	58
4.2.1 Experimentos.....	58
4.3 Comparación.....	64
5.CONCLUSIONES.....	68
6.REFERENCIAS Y BIBLIOGRAFÍA.....	72
ANEXOS.....	74
MANUAL DE INSTALACIÓN.....	74
MANUAL OPERACIÓN DEL SISTEMA.....	84

ÍNDICE DE TABLAS

Tabla	Página
2.1 Pseudocódigo del algoritmo de retropropagación.....	29
3.1 Percepciones de las presas.....	37
3.2 Movimientos de las presas.....	38
3.3 Percepciones del depredador.....	39
3.4 Movimientos del depredador.....	40
3.5 Pseudocódigo del depredador semi aleatorio.....	44
3.6 Pseudocódigo del depredador constante.....	45
3.7 Pseudocódigo del depredador aleatorio.....	45
3.8 Pseudocódigo del proceso de evolución.....	46
3.9 Pseudocódigo de la función de aptitud de evolución.....	47
3.10 Pseudocódigo del proceso de evolución con aprendizaje.....	49
3.11 Pseudocódigo de la función de aptitud de evolución con aprendizaje.....	50

ÍNDICE DE ILUSTRACIONES O FIGURAS

Ilustración	Página
2.1 Representación de los cromosomas antes del cruzamiento.....	19
2.2 Representación de los cromosomas después del cruzamiento.....	19
2.3 Representación de los cromosomas antes de la mutación.....	19
2.4 Representación de los cromosomas después de la mutación.....	20
2.5 Neurona.....	21
2.6 Sinapsis de una neurona.....	21
2.7 Potencial de una neurona.....	22
3.1 Ambiente del sistema.....	35
3.2 Representación de la malla de movimiento.....	36
3.3 Representación de percepción de las presas al norte.....	36
3.4 Representación de percepción de las presas al este.....	36
3.5 Representación de percepción de las presas al sur.....	36
3.6 Representación de percepción de las presas al oeste.....	37
3.7 Representación de los movimientos de las presas.....	38
3.8 Representación de percepción del depredador al norte.....	39
3.9 Representación de percepción del depredador al este.....	39
3.10 Representación de percepción del depredador al sur.....	39
3.11 Representación de percepción del depredador al oeste.....	39
3.12 Representación de los movimientos del depredador.....	40
3.13 Representación de una red neuronal de una presa.....	41
3.14 Representación interna de un peso de conexión.....	42
3.15 Representación interna de los pesos de conexión de una presa.....	43
3.16 Representación de la generación con los 1000 individuos.....	43
3.17 Representación del primer gen en el primer cromosoma.....	44
3.18 Representación de los cromosomas antes del cruzamiento.....	47
3.19 Representación de los cromosomas después del cruzamiento.....	48
3.20 Representación de los cromosomas antes de la mutación.....	48
3.21 Representación de los cromosomas después de la mutación.....	48

3.22 Representación de los cromosomas antes del cruzamiento.....	50
3.23 Representación de los cromosomas después del cruzamiento.....	50
3.24 Representación de los cromosomas antes de la mutación.....	51
3.25 Representación de los cromosomas después de la mutación.....	51
4.1 Evolución 1000 generaciones 100 individuos con depredador aleatorio.....	52
4.2 Simulación 1000 generaciones 100 individuos con depredador aleatorio.....	53
4.3 Evolución 1000 generaciones 1000 individuos con depredador constante.....	53
4.4 Simulación 1000 generaciones 1000 individuos con depredador constante.....	54
4.5 Evolución 1000 generaciones 1000 individuos con depredador semi aleatorio.....	54
4.6 Simulación 1000 generaciones 1000 con depredador semi aleatorio.....	55
4.7 Evolución 1000 generaciones 1000 individuos con depredador aleatorio y reducción de la malla toroidal.....	55
4.8 Simulación 1000 generaciones 1000 individuos con depredador aleatorio y reducción de la malla toroidal.....	56
4.9 Evolución 1000 generaciones 1000 individuos con depredador constante y reducción de la malla toroidal.....	56
4.10 Simulación 1000 generaciones 1000 individuos con depredador constante y reducción de la malla toroidal.....	57
4.11 Evolución 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de la malla toroidal.....	57
4.12 Simulación 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de la malla toroidal.....	58
4.13 Efecto Baldwin 1000 generaciones 100 individuos con depredador aleatorio.....	58
4.14 Simulación 1000 generaciones 100 individuos con depredador aleatorio.....	59
4.15 Efecto Baldwin 1000 generaciones 1000 individuos con depredador constante.....	59
4.16 Simulación 1000 generaciones 1000 con depredador constante.....	60
4.17 Efecto Baldwin 1000 generaciones 1000 individuos con depredador semi aleatorio.....	60
4.18 Simulación 1000 generaciones 1000 individuos con depredador semi aleatorio.....	61
4.19 Efecto Baldwin 1000 generaciones 1000 individuos con depredador aleatorio y reducción de la malla toroidal.....	61
4.20 Simulación 1000 generaciones 1000 individuos con depredador aleatorio y reducción de la malla toroidal.....	62

4.21 Efecto Baldwin 1000 generaciones 1000 individuos con depredador constante y reducción de la malla del toroidal.....	62
4.22 Simulación 1000 generaciones 1000 individuos con depredador constante y reducción de la malla del toroidal	63
4.23 Efecto Baldwin 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de la malla toroidal.....	63
4.24 Simulación 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de la malla toroidal.....	64
4.25 Evolución y efecto Baldwin 1000 generaciones 100 individuos con depredador aleatorio.....	64
4.26 Evolución y efecto Baldwin 1000 generaciones 1000 individuos con depredador aleatorio y reducción de la malla toroidal.....	65
4.27 Evolución y efecto Baldwin 1000 generaciones 1000 individuos con depredador constante y reducción de la malla toroidal	65
4.28 Evolución y efecto Baldwin 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de la malla toroidal.....	66
4.29 Pesos evolución y efecto Baldwin 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de la malla toroidal.....	66
4.30 Evolución y efecto Baldwin 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de la malla toroidal.....	67

1 INTRODUCCIÓN

1.1 Generalidades

Las computadoras electrónicas han revolucionado el desarrollo en la historia de la ciencia y la tecnología. Como resultado de esta transformación, se ha incrementado notablemente nuestra capacidad de predecir y controlar la naturaleza en formas que fueron raramente concebidas hace cien años. Los acontecimientos de esta revolución han derivado, entre otras cosas, en la creación de entidades “inteligentes” y formas alternativas de “vida” representadas en algoritmos y programas computacionales[1].

Los principales objetivos de crear inteligencia artificial y vida artificial tuvieron sus orígenes en los inicios de la era computacional. Los precursores de la ciencia de la computación Alan Turing, John Von Neumann, Norbert Wiener, entre otros, fueron motivados en gran parte por visiones de programas de computadora con inteligencia y capacidades semejantes a las naturales como auto-procrearse, capacidad de adaptarse y controlar su medio ambiente[1].

Estos investigadores, provenientes de disciplinas científicas consolidadas como la biología, la psicología y la electrónica, observaron los sistemas naturales como metáforas guía para conseguir sus visiones. Se intentó modelar el cerebro y aprendizaje humano, así como el proceso de la evolución. La metodología consistió en tomar aspectos de la biología y aplicarlos a los sistemas computacionales. Desde los inicios de la década de los 1980, se ha experimentado el renacimiento de toda una nueva corriente de investigación en ciencias computacionales basada en estos principios.

La inteligencia artificial y la vida artificial plantean problemas computacionales complejos que requieren de soluciones que resultan difíciles de programar artesanalmente. En sus inicios, las investigaciones en inteligencia artificial se fundamentaron en la premisa de que sería suficiente codificar reglas capaces de simular inteligencia en un programa; los sistemas expertos fueron un resultado de este optimismo. En cambio, ahora se cree que la mejor ruta a la inteligencia artificial es la creación de un modelo inductivo en el cual los humanos diseñan reglas simples y generan reglas complejas o inteligencia automáticamente por medio de la interacción con las mismas.

El conexionismo (computación inspirada en sistemas de neuronas) es un ejemplo de esta filosofía (Smolensky 1988). En esta teoría, las reglas están basadas en simulaciones de las neuronas, activaciones dispersas, reforzamiento y debilitación en las conexiones de las mismas; el comportamiento emergente es un patrón de comportamiento sofisticado que surge en una red neuronal durante la resolución de problemas como el reconocimiento de patrones y el aprendizaje[1].

Los algoritmos genéticos son otro ejemplo de estos sistemas. Estos fueron inventados por John Holland y fueron desarrollados por él mismo, sus estudiantes y colegas en la universidad de Michigan entre 1960 y 1970[1]. El objetivo principal de Holland no fue el diseñar algoritmos para

resolver problemas específicos, sino estudiar formalmente los fenómenos de adaptación como ocurren en la naturaleza y desarrollar formas en las cuales los mecanismos de adaptación natural pueden ser simulados en sistemas computacionales.

Los algoritmos genéticos son un método para transformar una población de cromosomas representados por cadenas binarias de longitud fija a una nueva población por medio de la selección natural y operadores inspirados en la genética como el cruce y la mutación. Cada cromosoma consiste en una colección de genes y cada gen es una instancia de un alelo o rasgo; en particular 0 o 1. El operador de selección toma aquellos cromosomas de la población que tendrán el derecho a reproducirse y en promedio, los más aptos producen más descendientes que los menos aptos. El cruce intercambia fragmentos de los cromosomas simbolizando una reproducción sexual entre dos organismos; la mutación altera aleatoriamente el valor de los rasgos en alguna posición específica del cromosoma[1].

Existen dos enfoques para el estudio de la evolución en los sistemas computacionales. El primero es encontrar algoritmos computacionales que permitan desarrollar mejores soluciones a problemas particulares y la segunda es poder entender fenómenos de adaptación por medio de modelos computacionales.

Desde hace algunas décadas, varios científicos computacionales han estudiado sistemas evolutivos con la idea de que la evolución puede utilizarse como una herramienta de optimización para resolver problemas de ingeniería[1]. La idea fundamental es evolucionar una población de soluciones candidatas a un problema, usando operadores inspirados en los mecanismos de variación del material genético de los seres vivos y la selección natural.

Para investigadores de computación evolutiva, los mecanismos de evolución resultan apropiados para problemas computacionales en diversas áreas, pues estos problemas requieren realizar búsquedas en un espacio muy grande de posibles soluciones. Un ejemplo de estos problemas computacionales es la ingeniería proteica computacional en la cual un algoritmo busca las secuencias de aminoácidos para encontrar propiedades específicas. Otro ejemplo, es buscar reglas o ecuaciones que pronostiquen las alzas y bajas del mercado financiero como la tasa de cambio del dinero.

Por otro lado, algunos problemas computacionales requieren de un programa de computadora que se adapte a la dinámica del medio ambiente. Esto se ejemplifica en problemas de como el control de robots, en los cuales el robot realiza una actividad en ambientes dinámicos y por medio de mecanismos computacionales debe adaptarse a las condiciones prevalecientes. Otros problemas necesitan que los programas computacionales sean innovadores para realizar tareas nuevas y originales, como realizar una actividad que no fue considerada en el diseño inicial del sistema.[22][3]

Por otro lado, las simulaciones computacionales pueden captar complicaciones que son difíciles de tratar como el estudio de los organismos y la interacción entre los mismos. Las computadoras de alta velocidad permiten a los científicos modelar sistemas adaptables y analizar el comportamiento de los mismos; lo cual sería imposible de realizar en un sustrato biológico.

A lo largo de la historia se han desarrollado varios modelos abstractos para explorar ideas particulares como selección sexual, evolución del aprendizaje y la interacción entre el aprendizaje y la evolución. Dichos modelos no hacen predicciones en particular que puedan ser falsificadas, sino que buscan explorar qué tan bueno puede ser un mecanismo particular para producir un comportamiento específico[1].

La principal motivación para la elaboración de este trabajo es estudiar las causas y efectos del comportamiento, evolución, aprendizaje en los seres vivos que habitan en ambientes simulados y tomar como base dicho estudio para entender de mejor manera estos fenómenos de adaptación y desarrollar de nuevos algoritmos o técnicas que permitan la solución de problemas que plantean la ciencia de la computación y la biología.

El incentivo sería estudiar preguntas como ¿Cómo influye el material genético en el comportamiento?, ¿Cómo influye el aprendizaje en el comportamiento? , ¿Qué es el efecto Baldwin?. ¿El aprendizaje guía la evolución? y formular hipótesis susceptibles de validarse por medio de simulaciones realizadas en sistemas computacionales.[12]

Este proyecto surge como el deseo de estudiar las bases genéticas del comportamiento con la finalidad de entender fenómenos biológicos como la evolución, el aprendizaje. Asimismo, este trabajo surge de la inquietud por encontrar soluciones computacionales a problemas complejos basadas en procesos de adaptación como es el aprendizaje, la evolución y la combinación de estos. De la misma manera, también se considera este trabajo como un proyecto interno de investigación que no tiene cláusulas de confidencialidad.

1.2 Problema planteado

En los estudios previos del efecto Baldwin existen limitaciones como es el no considerar un medio ambiente dinámico. Asimismo, la mayoría de los trabajos previos han fallado en analizar el comportamiento emergente y las implicaciones que resultan del mismo en el ambiente.[10]

El problema a tratar consiste en observar el tiempo de vida, las acciones y el comportamiento de las presas generados por el proceso evolutivo, el proceso evolutivo con aprendizaje y verificar la teoría de Baldwin en un simulador depredador presa artificial en el cual se considera un medio ambiente dinámico.

La importancia del problema radica en que es posible con determinar con base a estos estudios cuáles son los patrones de referencia que siguen los seres vivos para comportarse, así como inferir conclusiones sobre las causas, relaciones y efectos que tiene la evolución, el aprendizaje y el comportamiento en los seres vivos en un modelo computacional basado en el efecto Baldwin.

La importancia de este proyecto consiste en comprobar que existen modelos computacionales basados en técnicas de vida artificial que pueden ayudarnos a optimizar procesos de búsqueda o selección de acciones, así como al entendimiento de fenómenos biológicos como el comportamiento, el aprendizaje y la evolución.

1.3 Objetivos de la investigación e hipótesis

La hipótesis de este trabajo de tesis consiste en “explorar si el aprendizaje guía la evolución en un sistema de simulación depredador presa. También se quiere demostrar que este modelo puede ayudar al entendimiento de fenómenos biológicos y la posible generación de algoritmos basados en técnicas de vida artificial que contribuyan a resolver problemas computacionales complejos”.

Se eligió un modelo computacional puesto que éste promete más que comunicar un conocimiento existente, especialmente cuando se aplica en otras áreas, inclusive proporciona más que la contestación de preguntas de intereses interdisciplinarios, debido a que pueden catalizar o activar la integración de información desde el punto de vista de varias disciplinas. [21]

Además de proporcionar el lenguaje y los conceptos básicos, estos modelos permiten el intercambio directo entre participantes de diferentes disciplinas, incluyendo el principio de que éstos modelos por si mismos pueden ser combinados o compuestos en programas. Por ejemplo, una red de neuronal puede ser integrada a un algoritmo genético para modelar comportamientos de una población que evoluciona. El éxito de esta metodología depende de encontrar una correcta correspondencia al integrar modelos de la biología y la evolución, ya que deben tener características críticas en común o compartidas de sus modelos individuales. [21]

Se decidió utilizar un modelo depredador presa artificial debido a que en este ambiente la presa es un individuo que depende de la selección natural basada en principios biológicos como la “sobrevivencia del más apto”. Se eligió también para simular cómo una presa bajo presión de su medio ambiente o un depredador que lo pretende cazar puede sobrevivir o actuar, es decir, este modelo depredador presa permite fácilmente observar comportamiento y entender fenómenos emergentes dadas las premisas biológicas como aprendizaje y evolución. Finalmente se optó por esta relación ecológica porque es una de las más importantes en la naturaleza dado que establece el equilibrio de los ecosistemas. [5]

Un concepto fundamental en este trabajo es el comportamiento emergente que es aquel que resulta de la interacción entre los individuos y es complejo de modelar debido a que el comportamiento global no puede expresarse como combinación lineal de los comportamientos individuales. En el modelo depredador presa existen comportamientos de agrupación o dispersión de las presas para poder sobrevivir. Dichos comportamientos son emergentes porque no resultan de la suma de estrategias individuales prefabricadas sino simplemente surgen de la interacción entre las presas, de las presas con el depredador y con otros elementos de su medio ambiente.[15][12][24]

Otra razón por la cual se justifica este trabajo consiste en que, hasta donde sabemos, no existe un modelo computacional del efecto Baldwin en un modelo depredador presa en donde el ambiente juegue un papel importante y que muestre realmente la interacción entre la evolución y el aprendizaje para entender dichos fenómenos biológicos y su interacción.

1.4 Contribuciones esperadas

Se desea que este trabajo contribuya al entendimiento la relación depredador presa en el marco de los fenómenos de adaptación como es la evolución y el aprendizaje, así como el entendimiento del comportamiento de los agentes que participan en dicho modelo utilizando un enfoque sintético basado en simulaciones computacionales.

Adicionalmente, se busca comprobar que las técnicas computacionales utilizadas resultan apropiadas para trabajos futuros. Se espera que este trabajo ejemplifique como estas técnicas encuentran optimizaciones globales y no locales que se vayan evolucionando para encontrar inteligencia adaptativa y soluciones óptimas a problemas complejos como el depredador presa. También se pretende ampliar los estudios del efecto Baldwin en ambientes de vida artificial.

1.5 Organización del documento

El documento esta organizado en cinco capítulos. El primer capítulo es una introducción que muestra las generalidades, problema planteado, objetivos de la investigación, contribuciones esperadas, que motivan a la elaboración de este trabajo así como una breve explicación de la estructura de este documento. El segundo capítulo habla del marco teórico, vida artificial, teorías biológicas, técnicas computacionales y trabajos previos relacionados con el problema planteado. El tercer capítulo presenta el modelo propuesto, es decir, plantea el problema de manera más específica y las técnicas de cómo se pretende abordar el problema. El capítulo cuarto describe los experimentos y resultados obtenidos. El capítulo cinco son las conclusiones. Finalmente este trabajo presenta las referencias bibliográficas que se tomaron como base para la elaboración de este trabajo, así como el manual de instalación, operación y compilación de los códigos computacionales que fueron aval de esta tesis.

2 TRABAJOS RELACIONADOS

2.1 Vida artificial

La vida artificial integra diferentes disciplinas como las ciencias computacionales, la física, la biología, la economía y la filosofía. Uno de los fundadores, Christopher Langton, define la vida artificial como “una rama del conocimiento cuyo principal objetivo es entender la vida mediante la realización de la abstracción de principios dinámicos que se enfocan en fenómenos biológicos y recrean ese dinamismo en un ambiente, como las computadoras, para realizar nuevas formas de manipulación de experimentos y pruebas”. [21]

Mientras la biología, esencialmente analítica, busca descomponer fenómenos complejos con componentes básicos, la vida artificial es sintética pues busca construir fenómenos desde sus unidades más elementales. Con esta aseveración como premisa, la vida artificial complementa la investigación biológica a través de la exploración de nuevos caminos con la motivación de entender ese gran abismo y crucigrama que es la vida. [21]

El uso del término vida artificial significa que los sistemas en cuestión son hechos por humanos, es decir, los componentes básicos no fueron creados por la naturaleza a través de la evolución. Por ejemplo, John Von Neumann simuló un proceso reproductivo tan real como en la misma naturaleza. La diferencia consiste en los componentes básicos: las células artificiales y las células naturales. [21]

Los temas que surgen de cómo resultado de la investigación en vida artificial pertenecen a los fenómenos biológicos existentes como a los sistemas complejos en general. Es por ello, que la vida artificial persigue dos objetivos principales que son: incrementar nuestro entendimiento a la naturaleza y mejorar nuestra orientación hacia modelos artificiales y así proveernos con la habilidad de mejorar el desempeño de los mismos. Una ejemplificación del primer objetivo es el realizado por Von Neumann simulando la reproducción natural y un ejemplo del segundo objetivo es el desarrollo de software por medio de la evolución propuesto por John Koza. [15]

Otro aspecto de los sistemas con vida artificial es que consisten en una larga colección simple de unidades básicas cuyas propiedades más interesantes son aquellas que resultan a altos niveles. Parafraseando, los comportamientos emergentes o actitudes impredecibles que resultan de la interacción de pequeñas propiedades son una aportación importante que nos brinda la vida artificial. [3][21]

De acuerdo con Brooks, un estudio de la evolución de la vida en la tierra revela que la mayor parte del tiempo invertimos desarrollando inteligencia básica, es decir, la capacidad del individuo de acercarse o alejarse de algo o alguien de forma libre y apropiada, de imitar y de inhibir algo o a alguien que está a su alrededor [21]. La vida artificial nos ofrece la oportunidad de conducir experimentos que son extremadamente complicados para la biología y no visibles del todo fácilmente: complementa la biología pues provee herramientas para nuevos descubrimientos científicos.

2.2 Biología y evolución

2.2.1 Formas de adaptación en los sistemas biológicos

La evolución y el aprendizaje son dos formas biológicas de adaptación que difieren en espacio y tiempo. El primero es un proceso de reproducción selectiva y sustitución basada en la existencia de una población distribuida geográficamente de individuos que muestran alguna variedad. El segundo, en cambio, es un conjunto de cambios que suceden en el interior de cada individuo durante su existencia. Ambas formas biológicas operan en diferentes espacios del tiempo. [5][12]

La evolución es una forma de adaptación capaz de obtener relativamente cambios lentos en el ambiente que pueden abarcar varias generaciones; aprendizaje por otra parte permite que un individuo se adapte a los cambios ambientales que son impredecibles a nivel generación. El aprendizaje puede incluir una variedad de mecanismos que producen cambios de adecuación durante su existencia, como desarrollo físico, maduración de las neuronas, elasticidad de las articulaciones de los huesos. Mientras la evolución opera en el material genético, el aprendizaje opera solamente en las características obtenidas y los cambios en dichas características aprendidas o adquiridas no pueden afectar directamente el material genético. [13][9][5][12]

Existen tres teorías sobre la interacción entre el aprendizaje y la evolución. Al mismo tiempo, también existen modelos computacionales que han intentado simular dichas teorías. Inicialmente Lamarck surge con la idea que las características aprendidas son transmitidas completamente a los descendientes a través del material genético permitiendo así la evolución y la superación de los individuos en el tiempo. En otras palabras, afirma que todas las formas de vida surgen de un continuo proceso de modificación en la historia. Para explicar este proceso, se han aceptado teorías como la de características adquiridas, en las cuales las nuevas características en un organismo se desarrollan por una necesidad creada por el ambiente y transmitidas directamente a nuevas generaciones. [6]

La teoría de Darwin ha sido ampliamente reconocida como un hecho científico. En el modelo la sobrevivencia del más apto es relativamente simple. Dicha postura establece que la vida orgánica resulta de materia inorgánica a través de un proceso preestablecido en la tierra y que la forma original de vida va evolucionando hacia formas de vida más complejas a través de un proceso natural en el que intervienen cambios genéticos aleatorios y la selección natural. A diferencia de la teoría Lamarckiana, aquello que se aprendió no puede ser heredado a los descendientes. [5]

Otros aspectos importantes de la teoría de Darwin es la variación o diferencias entre varias especies, la herencia como la transmisión de información genética a los descendientes o futuras generaciones y la reproducción diferencial que consiste en que unas especies sobreviven de acuerdo a ciertas características heredadas. También existen modelos que simulan dicha teoría en donde su principal objetivo es modelar la selección natural en ambientes computacionales en donde el ambiente toma un papel importante en la evolución representada en sistemas computacionales. [5]

La tercera teoría surge el esfuerzo de explicar los límites de la evolución en registros fósiles. En 1896 surgió la idea de que la evolución puede ser influenciada por el aprendizaje durante la vida sin asumir que las características aprendidas directamente modifican el material genético. El argumento que Baldwin dio fue que el aprendizaje acelera la evolución porque individuos no aptos se pueden reproducir adquiriendo durante su vida características necesarias para vivir[7]. Dado que el aprendizaje necesita tiempo (pudiendo ser una desventaja), se muestra que la evolución tiende a seleccionar individuos que ya tienen de forma innata características que de otra forma serían aprendidas. [7][8]

Este último aspecto del efecto Baldwin, que renombra la asimilación genética indirecta de las características aprendidas, ha sido soportado por evidencia científica y definida por Waddington en 1942 como el efecto de canalización. La teoría de Baldwin se considera el punto medio entre la teoría Lamarckiana y la teoría Darwinista pues aquí las características aprendidas se transmiten indirectamente por asimilación genética. [7][8][9][13]

2.3 Computación

2.3.1 Algoritmos evolutivos

Esencialmente, un algoritmo evolutivo se basa en la obtención de conjuntos sucesivos de soluciones a un problema determinado, mediante procesos de muestreo aplicados a soluciones obtenidas mediante un mecanismo de *selección*. Por analogía con los procesos naturales de evolución de las especies, a cada conjunto de soluciones se le denomina *población* y a cada solución de una población se le denomina *individuo*. Los procesos más comunes de muestreo se denominan *cruce* y *mutación*, por esa misma analogía. Los mecanismos de selección, cruce y mutación suelen agruparse bajo la denominación común de *operadores*. Cada individuo tiene asociado un valor que indica la calidad del mismo, el cual se calcula usualmente evaluando la expresión matemática del problema a optimizar, este valor se suele denominar *función de aptitud*.

El proceso de optimización se mantiene hasta que se alcanza un criterio de terminación predefinido, el cual suele ser un predeterminado número de generaciones o que todas las soluciones de la población sean iguales. La optimización puede realizarse considerando una sola función (monoobjetivo) o sobre varias funciones (multiobjetivo) de aptitud. Entre los algoritmos evolutivos, se encuentran los denominados algoritmos genéticos (AG) que han sido empleados recientemente con éxito en problemas de optimización numérica y a problemas de optimización combinatoria. [1][14][18][19]

2.3.1.1 Algoritmo genético simple

En un algoritmo genético simple (AGS), los individuos son representados por medio de cadenas binarias. Si las variables del problema bajo estudio no son binarias, se procede a codificarlas en binario mediante procedimientos adecuados para esto. Todas las generaciones tienen un mismo número de individuos (tamaño) y entre los operadores comúnmente utilizados en la práctica cabe destacar: selección proporcional, cruce de un punto y mutación de un punto.

La **selección proporcional** se basa en asignar a cada individuo una probabilidad de selección que es proporcional al valor de su función objetivo, para esto se divide el valor de su función aptitud entre la suma de los valores de la función de aptitud de todos los individuos de la población. La selección se suele modelar mediante un mecanismo similar a una ruleta de casino, siendo el área asignada en la ruleta a cada individuo equivalente a su probabilidad de selección. En un AG simple se suelen seleccionar dos individuos, aunque existen procesos que emplean un mayor número de estos. Por analogía con los procesos biológicos, los individuos seleccionados se denominan *padres*.

El **cruce en un punto** se basa en la determinación aleatoria de un punto de cruce seguida del intercambio de fragmentos de las cadenas entre los individuos seleccionados. Para dar cierta probabilidad de supervivencia a los padres, ya que estos pueden ser muy buenas soluciones al problema tratado. El cruce está sujeto a cierta *probabilidad de cruce*, cuyos valores habituales oscilan alrededor de 0.6. Si el cruce no se realiza, entonces los padres son copiados directamente a la nueva generación. Por analogía con los procesos biológicos, los individuos resultantes del cruce se denominan *descendientes*. Ver figuras 2.1 y 2.2.

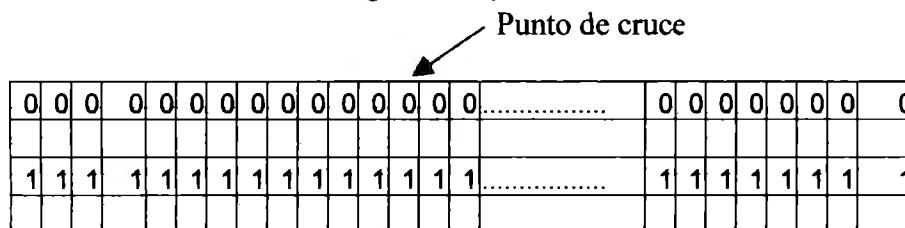


Fig. 2.1 Representación de los cromosomas antes del cruzamiento

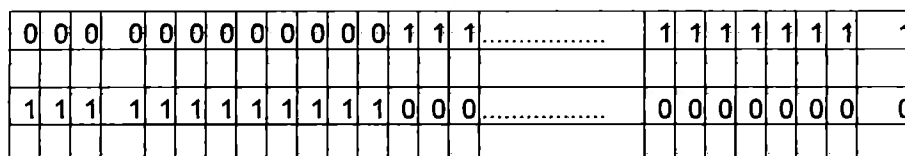


Fig. 2.2 Representación de los cromosomas después del cruzamiento.

La **mutación de un punto** consiste en cambiar el valor de una posición seleccionada al azar de entre todos los que componen los individuos resultantes de cruce. Al igual que el cruce, la mutación está gobernada por una *probabilidad de mutación*, de tal forma que solo existe una pequeña probabilidad de introducir mutaciones. Los valores típicos de dicha probabilidad oscilan alrededor de 0.001 en la práctica.

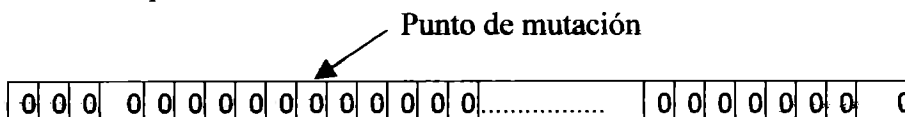


Fig. 2.3 Representación de los cromosomas antes de mutación

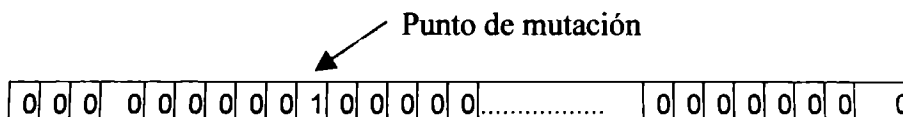


Fig. 2.4 Representación de los cromosomas después de la mutación.

La convergencia hacia la solución suele favorecerse considerablemente, en muchos tipos de problemas, mediante la preservación en poblaciones sucesivas de los mejores individuos encontrados hasta el momento. Esta técnica se denomina *elitismo*. Se han desarrollado otros muchos operadores de selección, cruce y mutación capaces de mejorar el desempeño de los algoritmos genéticos simples, así como sofisticados métodos de incrementar la convergencia hacia la solución óptima. También es posible implementar algoritmos genéticos en el dominio de los números reales. En la actualidad resulta difícil afirmar cuál de las dos codificaciones es superior, aunque varios autores concuerdan que ésta depende del problema bajo consideración. Es importante mencionar que en la actualidad existe una gran variedad de métodos avanzados en el uso de los algoritmos genéticos. [1][14][17][18][19]

2.3.2 Redes neuronales

En muchas aplicaciones del mundo real, se requiere que las computadoras resuelvan problemas complejos de reconocimiento de patrones. Debido a que las computadoras convencionales resultan inapropiadas para este tipo de problemas, se han empleado a las características de la fisiología del cerebro como la base para nuevos modelos de procesamiento. La tecnología se conoce como redes neuronales artificiales (RNA) o simplemente redes neuronales.

Una red neuronal artificial es una colección de procesadores en paralelo interconectados en la forma de un grafo dirigido, organizados de tal forma que se adapte al sistema o problema bajo consideración. Gráficamente, se puede representar cada elemento o unidad de procesamiento o unidad como un nodo, con conexiones entre unidades llamadas arcos. [14][18][17]

Existen al menos 2 modos de operación en las redes: modo de entrenamiento y modo de producción. El procesamiento en modo de entrenamiento de una red consiste en codificar la información del problema que se pretende solucionar. El proceso en modo de producción se refiere a presentar las entradas en la red para que calcule la solución al problema.

Haciendo una analogía con las neuronas biológicas podemos observar en la figura 2.3 y figura 2.4 que las dendritas son análogas a los arcos y el cuerpo neuronal análogo a la unidad de procesamiento en una computadora. Es importante describir el funcionamiento de una neurona natural para poder entender como funciona una neurona o unidad de procesamiento en el lenguaje computacional. [14][18][17]

193227

2.3.2.1 Fisiología de una neurona simple

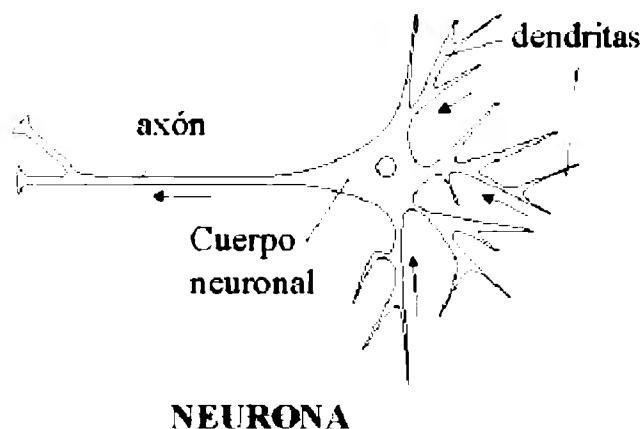


Fig. 2.5 Neurona

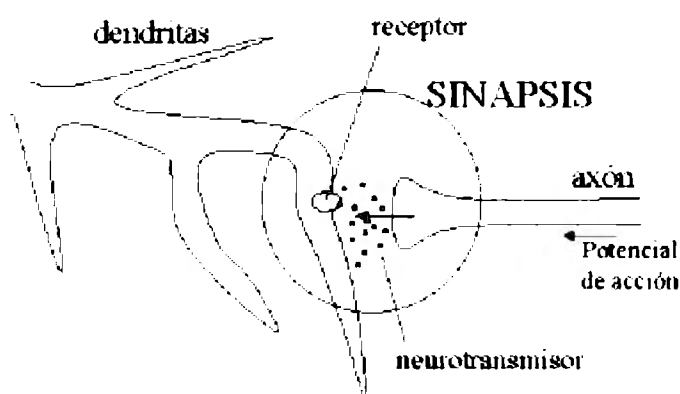


Fig. 2.6 Sinapsis de una neurona

Aquí se muestran los componentes de una célula nerviosa del sistema nervioso central en donde las estructuras principales de una célula típica son las dendritas, el cuerpo celular, el axón y las sinapsis, como se muestra en la figura. 2.6. Las dendritas tienen la tarea de transmitir activación de una neurona al cuerpo celular o neuronal. El cuerpo celular, a su vez, incorpora dicha activación y finalmente, el axón transmite la información a través de las sinapsis.

El axón de muchas neuronas se encuentra rodeado de una membrana llamada mielina. Los nodos de Ranvier son aquellos que interrumpen la mielina periódicamente en la longitud del axón. Las sinapsis conectan los axones de una neurona con varias partes de otra neurona. [18]

La membrana de una neurona separa el plasma intracelular del líquido externo en la célula. La membrana es permeable cierto tipo de especies iónicas y actúa para mantener el diferencial de potencial entre el fluido intracelular y el fluido extracelular. Permite también a través de bombear sodio y potasio. El mecanismo transporta iones de potasio a la célula. Otras especies iónicas presentan iones de cloro y iones orgánicos negativos.

Todas las especies iónicas pueden difundirse a través de la membrana celular, a excepción de los iones orgánicos, ya que resultan ser extremadamente largos. Dado que los iones orgánicos no pueden difundirse hacia afuera de la célula, la carga neta negativa provoca que la difusión de cloro en la célula sea desfavorable pues existe una mayor concentración de iones de cloro fuera de la célula. Los impulsos de sodio y potasio obligan a que exista una mayor concentración de potasio dentro de la célula y una mayor concentración de sodio fuera de la célula. [17][18]

La membrana celular es selectivamente más permeable a los iones de potasio que a los iones de sodio. El gradiente químico de potasio tiende a ocasionar que los iones de potasio se desplacen hacia el exterior de la célula, sin embargo la fuerte atracción de los iones orgánicos tiende a mantener el potasio en el interior de la célula. El resultado de las fuerzas que se oponen es que el equilibrio se consigue en el caso en donde existen significativamente más iones de sodio y cloro fuera de la célula, y más potasio y iones orgánicos dentro de la célula.

El equilibrio obtenido provoca un diferencial de potencial a través de la membrana celular de 70 a 100 milivolts, con un fluido intracelular que es más negativo. El potencial es llamado potencial de reposo y se muestra en la figura 2.7:

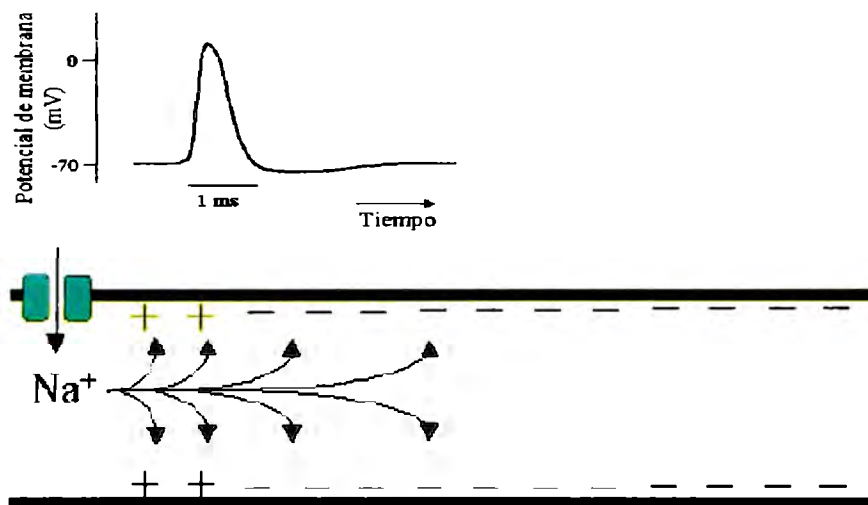


Fig. 2.7 Potencial de una neurona

Las entradas que excitan a la célula reducen el diferencial de potencial en la membrana celular. La separación de polos resultante en los axones altera la permeabilidad de los iones de sodio en la membrana celular. Como resultado de esto, existe un flujo de iones de sodio positivos en la célula contribuyendo a la despolarización global. En consecuencia, se genera un efecto en la acción de potencial.

Las fibras nerviosas por sí mismas son conductores pobres. La transmisión de acción de potencial hacia el axón es el resultado de una secuencia de despolarizaciones que ocurren en los nodos de Ranvier. Cuando un nodo se despolariza, este activa la despolarización en el siguiente nodo. La acción de potencial viaja a través de las fibras en una forma discontinua de un nodo a

otro. Una vez que la acción de potencial ha rebasado un punto, éste será incapaz de ser excitado en un milisegundo, mientras se reestablece a su potencial de resistencia. Este periodo de refacción implica una limitación en la frecuencia de transmisión de impulsos nerviosos de 1000 por segundo. [17][18]

2.3.2.2 Conexiones sinápticas

La actividad que existe en la conexión entre dos neuronas llamada sinapsis puede verse como la comunicación entre neuronas que resulta de una liberación de una célula presináptica de sustancias llamadas neurotransmisores y que son sustancias absorbidas por la célula postsináptica. Cuando la acción de potencial llega a la membrana presináptica, los cambios en cambia la permeabilidad de la membrana causan el influjo de iones de calcio. Estos iones causan que los versículos que contienen neurotransmisores se fusionen con la membrana presináptica y que sean liberados a la fisura sináptica. (Ver figura 2.6.)

2.3.2.3 Redes neuronales artificiales

En redes neuronales artificiales existen cinco aspectos esenciales:

a) Características del nodo.

En una red neuronal, se requiere especificar cómo la activación entrante se suma y se procesa para generar el nivel de activación y cómo se genera la salida; esto es, qué se necesita especificar las características del nodo. Las funciones activación más conocidas son:

- La función lineal: simplemente suma las entradas. La neurona se activa hasta que el umbral es alcanzado.
- La función escalón: también llamada función de umbral lineal
- La función sigmoideal o logística: es en esencia una versión de la función escalón. Su valor es cero o menos uno para el valor más bajo. Al mismo tiempo va creciendo rápidamente y a altos niveles en la entrada se satura. Esta propiedad de saturación puede observarse en la naturaleza, en los casos en los cuales los factores biológicos saturan el nivel de activación de las neuronas[18]. Cabe mencionar que la pendiente de la función sigmoideal puede ser modificada.

b) Conectividad

La segunda propiedad a especificar en el diseño de una red neuronal consiste en observar como los nodos individuales son conectados entre sí. Esto puede estar representado con nodos y arcos en un grafo dirigido. Las conexiones son en una sola dirección; si son bidireccionales deberán ser especificadas con dos flechas. A las redes neuronales que tienen dos direcciones se les conoce como redes recurrentes. [17]

Al menos existen tres tipos de capas por definir:

- Una capa de entrada
- Cero o más capas intermedias u ocultas
- Una capa de Salida

Para redes grandes, la notación de grafos se vuelve difícil por lo que generalmente se utilizan matrices. La idea es listar todos los nodos horizontalmente y verticalmente. Los elementos que se encuentran en las casillas de la matriz representan los pesos de conexión denotados W_{ij} que significa que el nodo j está conectado con el nodo i con un peso de conexión en particular o peso W . A dicha matriz se le conoce como matriz de conexión.

Los pesos de conexión determinan que tanta activación es transferida de un nodo al siguiente. Las conexiones positivas son excitatorias en lo que las conexiones negativas son inhibitorias. El cero significa que no existe alguna conexión entre nodos. Realizando una analogía con las redes neuronales biológicas, los pesos de conexión son también llamados pesos sinápticos. Los pesos son típicamente ajustados por una regla de aprendizaje hasta que la red es capaz de realizar una tarea en particular.

Al igual que en el álgebra lineal, el término vector es utilizado frecuentemente en la terminología de las redes neuronales. A los valores de los nodos de entrada se le llaman vector de entrada. similarmente a la lista de valores de activación de la capa de salida se conoce como vector de salida.

Las redes neuronales son clasificadas respecto a su conectividad. Si la matriz de conexiones tiene puros ceros en la diagonal se la llaman redes de alimentación hacia delante dado que solo hay conexiones hacia delante (feedforward), es decir, no existen ciclos pues son en una sola dirección. Una red con varias capas conectadas de forma hacia delante se le conoce como redes multicapas con alimentación hacia delante o perceptrón multicapas.

Si todos los nodos de una capa se conectan a todos los nodos de la siguiente capa se dice que las dos capas están totalmente conectadas. Las redes en las cuales los nodos están conectadas entre ellas en ambas direcciones se les llama redes de Hopfield. [17]

c) Reglas de propagación

La tercera propiedad especifica la forma en que la activación es propagada a través de la red. Normalmente, se considera una suma ponderada para determinar la propagación. En esta parte se necesita especificar claramente dicha regla, ya que si intentamos modelar neuronas, se necesita tener en cuenta las propiedades temporales de propagación, así como tiempos de retraso de la misma.

d) Reglas de aprendizaje

Las reglas de aprendizaje se requieren en redes neuronales para modificar los pesos. La tasa de aprendizaje indica que tan rápido cambian los pesos. Se considera que no debe ser muy pequeña porque sino el aprendizaje sería lento ni tampoco muy grande porque entonces la red reaccionaría muy fuertemente a fluctuaciones en el ambiente y no se estabilizaría fácilmente. Los pesos cambian de la forma siguiente:

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}$$

En donde t representa el tiempo. Aplicando esta fórmula se observa que el peso en el tiempo siguiente depende del valor del peso anterior.

e) Inserción de la red a un agente

Si queremos utilizar una red neuronal para controlar el comportamiento de agentes autónomos se requiere especificar como las señales sensoriales van a influenciar a la red y cómo los cálculos en la red van a influenciar en el comportamiento del agente. En otras palabras, debemos conocer las características de los sensores y el sistema motriz.

Algunos tipos de redes neuronales artificiales son:

- **Redes neuronales con aprendizaje por refuerzo:** Si un maestro le dice al estudiante que está mal o bien pero le deja la tarea de determinar porque esta bien o mal es una analogía a lo que significa aprendizaje por refuerzo. Al problema de atribuir un error o éxito se le conoce como el problema de asignación de créditos o castigos y es fundamental para teorías de aprendizaje. El término aprendizaje por refuerzo tiene un significado más técnico que es utilizado en la literatura de redes neuronales, esa definición es asignar aprendizaje en el cual un comportamiento en particular debe ser reforzado. Parafraseando, se recibe una señal de refuerzo positivo si el resultado que ha producido es exitoso, una señal de refuerzo nulo o una señal de refuerzo negativo si no ha sido exitoso. [17]
- **No supervisadas:** Se le llaman así a las redes neuronales que no necesitan que se indique cuál es el resultado correcto esperado de la red.
- **supervisadas:** El término supervisado en el sentido estricto significa que para ciertas entradas, las salidas correspondientes se conocen. En este caso, la salida correcta debe conocerse por el algoritmo de aprendizaje. Las redes solamente requieren aprender el mapeo de las entradas con las salidas predefinidas y el cambio de los pesos depende de la magnitud del error que una red produce en la capa de salida. Mientras mayor sea el error, mayor es la diferencia entre la salida que la red produce y el valor correcto de la salida, por lo tanto, el peso debe cambiar más. El algoritmo más común es retropropagación y el potencial para aplicaciones es muy amplio. Dicho algoritmo tiene limitaciones pues cae en mínimos o máximos locales, lo cual no garantiza una optimización global, no extrapola bien, y es lento. [18]

2.3.2.4 Retropropagación (“back propagation”)

Existen aplicaciones computacionales que son difíciles de implementar porque existen muchos problemas asociados a la solución basados en un proceso secuencial. Las aplicaciones deben realizar una traducción de datos compleja y en la actualidad no existe una función que obtenga la correlación uno a uno para describir el proceso de traducción. En sistemas de redes neuronales artificiales se ha encontrado un método para problemas que requieren soluciones de reconocimiento de patrones complejos y realización de funciones de correlación uno a uno. Este método es llamado retropropagación o “back propagation”. [20]

Retropropagación es un proceso formalizado a principios de los 70's por Paul Werbos. Después por Parker, Rummelhart y McClelland. La composición de la red es multicapa, con alimentación hacia delante utilizando un aprendizaje de modo supervisado. Parafraseando, es un sistema es capaz de centrarse en características de una entrada arbitraria que se parecen a otros patrones ya vistos previamente. [17]

En principio la red aprende un conjunto de entradas y salidas predefinidas utilizando dos fases del ciclo de adaptación. Después de que un patrón de entrada se aplica como un estímulo en la primera capa o capa de entrada, este se propaga a través de las capas siguientes hasta que se genera una salida.

El patrón de salida es entonces comparado con la salida deseada y una señal de error se procesa para cada neurona de salida. Las señales de error que son transmitidas hacia atrás, es decir, desde la capa de salida a cada nodo en las capas intermedias que contribuyen directamente con la salida. De todas formas, cada nodo en la capa intermedia recibe solamente una parte de la señal de error total, basada rigurosamente en la contribución relativa que hizo cada nodo a la salida original.

El proceso se repite, capa por capa, hasta que cada nodo en la red ha recibido una señal de error que describe su contribución relativa al error total. Basado en la señal de error recibida, la conexión de los pesos se actualiza por cada nodo para causar que la red haga convergencia hacia un estado que permita que todos los patrones de entrenamiento estén codificados o relacionado: uno a uno.

La importancia de este proceso es que al mismo tiempo que la red neuronal se entrena, los nodos en las capas intermedias se organizan de tal forma que todos aprenden a reconocer diferentes características del espacio total de entradas. Después de entrenar la red, cuando se presenta un patrón arbitrario incompleto o con ruido, las neuronas en las capas intermedias de la red responden con una salida activa solamente si la nueva entrada contiene un patrón que es similar a las características que una neurona entrenada ya aprendió a reconocer durante el entrenamiento. Asimismo, las neuronas de las capas intermedias tienen una tendencia a inhibir aquellas salidas si el patrón de entrada no contiene la característica que fue entrenada para reconocer. [20]

Como las señales se propagan a través de diferentes tipos de capas en la red, la actividad de un patrón en cada capa superior puede concebirse como un patrón con características que deben ser reconocidas por neuronas en las capas subsecuentes. El patrón de salida generado

puede ser interpretado como una correlación de la característica que nos proporciona la información en presencia o ausencia de muchas combinaciones de características en la entrada.

El efecto total de este comportamiento es que la red de retropropagación proporciona un modelo acerca de cómo permitir que un sistema computacional examine patrones de datos que puedan estar incompletos o con ruido y que reconozca patrones de una entrada parcial.

Muchos investigadores han demostrado que durante el entrenamiento las redes que utilizan retropropagación tienden a desarrollar relaciones internas entre nodos de tal forma que se organizan los datos de entrenamiento en clases de patrones. Esta tendencia puede ser extrapolada a la hipótesis de que todas las neuronas de las capas intermedias en una red que utiliza retropropagación están de alguna manera asociadas a una característica específica de los patrones de entrada como un resultado del entrenamiento. El tipo de asociación puede ser o no evidente a observador humano. [20]

Lo que sí es importante es que la red ha encontrado una representación interna que le permite generar las salidas deseadas dados los patrones de entrada de entrenamiento. Esta misma representación interna puede ser aplicada a entradas que no son utilizadas durante el entrenamiento. La red de retropropagación clasificará previamente estas entradas no vistas de acuerdo a las características que se comparten con las propiedades de los ejemplos de entrenamiento.

En conclusión retropropagación es un algoritmo que su nombre se debe a que la modificación de los pesos esta dictada por las reglas de aprendizaje dado que se propaga hacia atrás de la capa de entrada hacia la capa de salida. De hecho, el algoritmo puede ser observado intuitivamente como una serie de ondas que van hacia delante y hacia atrás en una cierta actividad. En la fase hacia delante la red produce las salidas dados los pares entrada-salida.

En la fase de hacia atrás, los pesos de las neuronas de salida y luego aquellos pesos de las neuronas intermedias regresan hacia las entradas y son modificadas capa por capa de acuerdo con la regla de aprendizaje y la intención de reducir el error. Los detalles del algoritmo son simple matemática y formalmente el algoritmo en:

Suponiendo que para una red de tres capas tenemos:

$$o_k = \frac{1}{1 + e^{-net_k}}$$

$$net_k = \sum_j w_{jk} o_j$$

$$o_j = \frac{1}{1 + e^{-net_j}}$$

$$\text{net}_j = \sum_i w_{ij} o_i$$

Retropropagación se deriva asumiendo que se quiere minimizar el error en las neuronas de salida para todos los patrones usando la siguiente fórmula de error:

$$E = \frac{1}{2} \sum_p \left(\sum_k (t_{pk} - o_{pk})^2 \right)$$

donde p es el número de patrones, k es el número de neuronas de salida, T_{pk} es la salida deseada y O_{pk} es la salida producida por la red en el patrón p de la salida k entonces la primera parte del problema es encontrar como E cambia como W_{ij} para esto obtenemos las siguientes derivadas:

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial \text{net}_k} \frac{\partial \text{net}_k}{\partial w_{jk}} = -(t_k - o_k) o_k (1 - o_k) o_j$$

en donde:

$$\delta_k = (t_k - o_k) o_k (1 - o_k)$$

$$w_{jk} \leftarrow w_{jk} + (-\eta) (-\delta_k) o_j (1 - o_j) o_i$$

La fórmula de los cambios en los pesos en la capa intermedia que conecta con la salida es:

$$\Delta w_{jk} = \eta \delta_k o_j$$

Ahora la segunda parte del problema es ver como afectan los pesos que conectan la capa intermedia con la capa de entrada y la regla de la cadena produce lo siguiente:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \sum_k \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial \text{net}_k} \frac{\partial \text{net}_k}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial w_{ij}} \\ &= \sum_k -(t_k - o_k) o_k (1 - o_k) w_{jk} o_j (1 - o_j) o_i \\ &= \sum_k -\delta_k w_{jk} o_j (1 - o_j) o_i \\ &= -o_i o_j (1 - o_j) \sum_k \delta_k w_{jk} \end{aligned}$$

lo que lleva a que la fórmula de los cambios en los pesos en la capa intermedia que conecta con la capa de entrada sea:

$$\Delta w_{ij} = \eta \delta_j o_i$$

Finalmente existe una regla que se puede generalizar para redes de más de tres capas y es:

$$\Delta w_{ij} = \eta \delta_j o_i$$

que especifica los cambios en los pesos que tienen las neuronas que conectan la capa de entrada con las n capas intermedias y luego:

$$\delta_j = (t_j - o_j) o_j (1 - o_j)$$

especifica la señal de error para la capa de salida y finalmente resulta:

$$\delta_j = o_j (1 - o_j) \sum_k w_{jk} \delta_k$$

que describe la señal de error para la unidad intermedia que conecta con la capa de salida. [23]

En resumen el procedimiento básico consiste en la siguiente tabla:

Inicializar los pesos
Presentar un vector de patrones de entrada a la red
Evaluar las salidas de la red propagando las señales hacia delante
Para todas las salidas de neuronas calcular la diferencia entre la salida deseada de la neurona con la salida actual de dicha neurona asumiendo una función de activación sigmoideal.
Para todas las demás neuronas desde las capas escondidas o intermedias hasta la capa de entradas calcular la capa subsecuente
Actualizar los pesos de acuerdo a las reglas de aprendizaje y algoritmo de aprendizaje.
Repetir todos los pasos un cierto número de iteraciones hasta llegar al menor error posible que se compara con el valor predefinido

Tabla 2.1 Pseudocódigo del algoritmo de retropropagación

Las redes neuronales, que utilizan como método de aprendizaje retropropagación, son una herramienta versátil y flexible pues son aplicadas a un número muy grande de problemas. En otras palabras, su versatilidad se debe a la naturaleza generalizada del procedimiento de aprendizaje de la red. [17]

Como ya se observó existen dos ecuaciones que se necesitan para retro propagar las señales de error dentro de la red. Ambas son utilizadas dependiendo de que neurona de procesamiento recibe la señal de error que contribuye directamente a la salida. Aquellas unidades que no se conectan directamente con la salida utilizan el mismo mecanismo de propagación del error sin importar la estructura de la red.

La generalización ofrece por este proceso común un arreglo y una conexión en las neuronas individuales dentro de la red. Cabe señalar que pueden cambiar bastante. Del mismo modo debido a la variedad de las estructuras en la red que se pueden crear y entrenar de forma exitosa utilizando los algoritmos de retropropagación, este aprendizaje aplicado a las redes puede ser aplicado a muchos problemas como compresión de datos, reconocimiento de patrones. [20]

2.3.3 Evolución de redes neuronales artificiales

La evolución ha sido introducida a las redes neuronales en tres rigurosos niveles diferentes: conexión de pesos, arquitecturas y reglas de aprendizaje. La evolución de conexión de pesos introduce un acercamiento adaptable y global al entrenamiento, especialmente en aprendizaje por refuerzo y en el paradigma de aprendizaje recurrente en redes neuronales en las cuales los algoritmos de entrenamiento basados en el algoritmo de retropropagación por lo regular han experimentado muchas dificultades. La evolución de arquitecturas permite en las redes neuronales adaptar topologías para diferentes tareas sin la intervención humana, por lo que proporciona un acercamiento al diseño automatizado tanto en los pesos de conexión, como en las estructuras ya que pueden ser evolucionadas

La evolución de reglas de aprendizaje pueden observarse como el proceso de aprender a aprender en una red en donde la adaptación de las reglas se consigue a través de la evolución. También puede observarse como un proceso adaptativo del descubrimiento automático de nuevas reglas. [14][17]

2.3.3.1 Evolución de pesos de conexión.

El entrenamiento de los pesos en una red neuronal está usualmente propuesto como la minimización de la función de error, como por ejemplo el error cuadrático medio. La mayoría de algoritmos de entrenamiento supervisado se basan en gradiente descendiente.

El problema es consiste en que durante el entrenamiento de una red se alcanza un mínimo local en la función de error, por lo que ésta es incapaz de encontrar un mínimo global. Esto ocurre cuando dicha función es multimodal o no diferenciable. La evolución artificial puede ser utilizada efectivamente en el entrenamiento de redes neuronales para encontrar el más cercano y óptimo conjunto global de pesos de conexión.

2.4 Trabajos previos

Hinton y Nowlan [10] han proporcionado un modelo computacional simple que demuestra cómo el aprendizaje puede ayudar y guiar la evolución (efecto Baldwin). Los autores consideraron un caso en el cual una red neuronal de un organismo tiene aptitud reproductiva sólo si está conectada correctamente. En el peor caso, no existe un camino para la red correcta y solamente una búsqueda evolutiva puede descubrir cuál de las conexiones potenciales debe estar presente tratando las posibilidades aleatoriamente. Inclusive, ellos consideran que la red óptima

es como una aguja en un pajar. Hinton y Nowlan[10] establecen: "... Es como buscar una aguja en un pajar cuando alguien te dice que estás cerca".

En sus experimentos, los individuos consisten de un material genético representado por veinte genes que codifican una red neuronal de veinte conexiones potenciales. Los genes pueden tener tres valores: cero, uno y signo de interrogación, que representan, respectivamente, presencia de la conexión, ausencia de la conexión y un estado cambiante. El mecanismo de aprendizaje consiste en un proceso aleatorio simple que intercambia en los genes el signo de interrogación por un cero o un uno. Esto hace los pesos en las conexiones se modifiquen hasta que se encuentra alguna combinación óptima durante la vida limitada del individuo. [10]

Este modelo sencillo que ejemplifica el efecto Baldwin, se basa en que los caracteres que están inicialmente adquiridos por medio del aprendizaje pueden ser después compuestos en el material genético. Una vez que los individuos tienen parte de sus genes con los valores correctos, los valores no especificados o aprendidos son seleccionados y los individuos con menos genes aprendidos tienden a ser seleccionados dado que la aptitud incrementa proporcionalmente. Como resultado decrece el número de genes que aprenden. En otras palabras, las características que primero fueron adquiridas a través del aprendizaje tienden posteriormente a codificarse genéticamente.

Por ejemplo, en el caso de Hinton y Nowlan existe la necesidad de regular un comportamiento óptimo por lo que el uso de un tipo de aprendizaje resulta conveniente. El problema de adquirir un lenguaje es otro ejemplo, pues resulta difícil imaginar que teniendo la habilidad de comunicarse en algún idioma en particular pueda ser codificado en nuestro material genético, en lugar de ser aprendido.

Una limitación del trabajo de Hinton y Nowlan, los primeros en desarrollar un modelo computacional del efecto Baldwin, es que el aprendizaje está representado por cambios aleatorios. En la vida real el aprendizaje depende de diversas variables. Por ejemplo, los cambios fenotípicos que experimentan los seres vivos como respuesta de adaptación a un medio ambiente adverso.

Otra debilidad del modelo de Hinton y Nowlan es que las simulaciones no tienen un ambiente independiente en el cual los organismos viven, evolucionan y aprenden. En cambio, es importante reconocer para entender fenómenos biológicos que la evolución y el aprendizaje resultan de la interacción entre organismos y ambientes. La evolución basada en aptitud no depende solamente en el material genético, sino que también en el ambiente.

Otro punto a considerar que motiva a proponer un modelo alternativo del efecto Baldwin es que en las simulaciones de Hinton y Nowlan, el proceso evolutivo y el proceso de aprendizaje determina las posibilidades de selección de un individuo en particular. El desempeño está en el proceso de aprendizaje y los individuos se reproducen si encuentran una pronta solución a los procesos de aprendizaje en su vida. Por esta razón, se puede inferir que la aptitud ligada o basada en el desempeño de aprendizaje es lo que explica cómo el aprendizaje guía la evolución en ciertas simulaciones.

En organismos reales no está definido que un buen desempeño en el aprendizaje automáticamente se traduce en alta aptitud. Los procesos de aprendizaje evolucionan y aparentemente convergen con la aptitud pero especialmente en organismos más avanzados la relación entre los dos puede ser totalmente compleja e indirecta. [4]

Finalmente, desde la perspectiva de la simulación, un punto débil de este trabajo es que el ambiente no cambia. Dado que en la vida real nos encontramos en un ambiente dinámico en el que los factores externos influyen en el comportamiento de los individuos, consideramos que resulta indispensable agregar este elemento en las simulaciones.

Existen varios modelos computacionales basados en la postura de Lamarck como el presentado por Ackley y Littman [13]. En este modelo se simulan los cambios aprendidos como resultado de una herencia lamarckiana; en su modelo simulan y analizan el cambio de un gen durante el proceso evolutivo. También existen otros desarrollos informáticos que siguen dicha teoría. Sin embargo, aunque esta hipótesis ha sido motivo de controversia, finalmente fue rechazada por los principios establecidos de la herencia, pero sirvió para que Charles Darwin trabajara con la influencia de ambiente en los procesos evolutivos.

Integrando estas ideas, Michael Littman muestra ejemplos de cómo el aprendizaje y la evolución se combinan en simulaciones computacionales. Inclusive él se cuestiona acerca de la relevancia entre la biología y la computación, es decir, ¿qué problemas pueden ser resueltos por aprendizaje?, ¿qué otros problemas pueden ser resueltos por evolución? y ¿cuáles necesitan la combinación de ambos?. Mucha gente ha hecho analogías entre aprendizaje y evolución, inclusive algunos han creído que el aprendizaje es un tipo de evolución. Por ejemplo, algoritmos que evolucionan o técnicas inspiradas en evolución que se consideran como parte del proceso de aprendizaje de máquina. [13]

¿Qué problemas se pueden resolver con la evolución, el aprendizaje o ambos?.

Para responder a dichos cuestionamientos, Littman define evolución como “cambios en el genoma que alteran el comportamiento innato” y aprendizaje como “cambios en el comportamiento del individuo como el resultado de la interacción con su ambiente”. La palabra “comportamiento” se refiere a cambios en las características del individuo.

En los modelos de Littman, los problemas no pueden ser solucionados por evolución porque el ambiente cambia tan rápido que no puede ser rastreado por el lento proceso de evolución genética o es tan complejo, que resulta imposible para la evolución codificarlo genéticamente.

Finalmente, Littman se combinó con Ackley para llegar a una idea que es el aprendizaje reforzado por la evolución, en el cual la asignación de ciertas acciones como placer y dolor pueden evolucionar en una arquitectura de redes neuronales. Este modelo se aproxima al aprendizaje de máquina que combina la evolución y el aprendizaje, en que tiene potencial para ser más poderosa que otras técnicas estándares convencionales[13].

Una limitación de Littman es que cae en un aspecto aleatorio tan alto que no puede seguir el proceso evolutivo lo cual provoca que las acciones sean al azar en lugar de un procedimiento que garantiza convergencia en la optimización.

Otra limitación de Littman y Ackley es que se basan en un algoritmo natural que fue descartado por la teoría darwinista lo que hace que ellos se alejen de la posibilidad de simular un ambiente natural realista.

Nolfi y sus colegas han investigado la interacción entre aprendizaje y evolución en donde las tareas de aprendizaje se diferencian de las tareas evolutivas [3][9]. El modelo consiste en un caso de agentes artificiales conocidos como “animats” que evolucionan en un nivel poblacional y aprenden en un nivel individual. En particular, los individuos que fueron seleccionados debido a la habilidad de obtener alimento en su ambiente también aprendieron a predecir las consecuencias sensoriales de sus acciones motrices en su vida. Cabe resaltar que Nolfi implementa una supervisión necesaria para que esta tarea sobre el aprendizaje sea flexible al ambiente en donde se encuentra. Profundizando, cada animat vive en un ambiente de una malla bidimensional donde existen varios puntos que representan comida y distribuidos aleatoriamente. Cada porción de alimento ocupa una celda; Si el animat se ubica en dicha celda se considera que el alimento se ingiere automáticamente y la aptitud del animat aumentada. Los individuos están equipados con una red neuronal conectada a un sistema sensomotriz que proporciona información de entrada en distancia y ángulo (respecto al animat) del alimento más cercano.

En este sistema, dos entradas codifican la distancia y el ángulo a la siguiente comida tomada y dos unidades codifican las cuatro posibles acciones: moverse derecha noventa grados, moverse izquierda noventa grados, moverse una celda adelante y mantenerse parado. En cada paso, la red neuronal recibe la entrada como una información sensorial de la comida más cercana, planifica una acción y produce una salida que consiste en la siguiente acción planeada, y la predicción del siguiente estado después del movimiento planeado.

En ese momento la acción planeada usada como entrada es ejecutada y la siguiente acción planeada es pasada como entrada; la nueva información sensorial es utilizada como entrada y señal de enseñanza para las unidades que codifican las salidas. Este modelo utiliza un método que compara y ajusta los pesos de las conexiones entre las cuatro entradas, las siete neuronas escondidas y las dos unidades de predicción en la salida con el algoritmo retropropagación

Los resultados muestran que después de varias generaciones, los individuos aprenden a predecir y aumentar su habilidad de encontrar comida en el tiempo. Comparando los resultados con experimentos en donde el aprendizaje no está permitido, se ha demostrado que las poblaciones que aprenden despliegan rápidos y altos valores de aptitud a lo largo de las generaciones que aquellas poblaciones que no aprenden. Dichos resultados han sido obtenidos también por Parisi y Harvey[9][3].

La explicación de Hinton y Nowlan no es suficiente para explicar estos resultados, es decir, una nueva explicación entre la interacción entre el aprendizaje y la evolución fue propuesta por Nolfi basada en el aprendizaje por refuerzo del ambiente.

Una limitación del modelo de Nolfi es que trata la evolución y el aprendizaje como dos procesos separados, es decir, no observa la sinergia que puede brindar un algoritmo que integre ambas.

Otra limitación del modelo de Nolfi es que utiliza el algoritmo retropropagación que aunque es el más utilizado en la práctica, tiene limitaciones tales como la localización de mínimos locales, y una complejidad computacional alta $O(n^3)$, además de ser un algoritmo supervisado, lo cual requiere de la especificación de las respuestas correctas, lo cual puede ser extremadamente complicado en la práctica.

Craig Reynolds[4] realizó un proyecto que emplea un modelo depredador presa pero en el cual se utiliza la programación genética para simular la selección de acciones de la presa. En este modelo computacional que simula a depredador y presas se describe un ambiente rectangular en donde existen obstáculos que hay que evadir. Usa programación genética y la función de aptitud es la suma de las presas. Agrega funciones específicas de comportamiento como ir a donde está un amigo, buscar obstáculos, buscar un depredador.

Una limitación de Craig Reynolds es que el comportamiento emergente en su simulación resulta ser no tan emergente, pues él predefine los comportamientos como acciones de las presas, como parte del sistema y pues nosotros vimos que el comportamiento emergente son acciones que surgen de estrategias sencillas no contempladas, es decir, debe resultar de la interacción, de acciones básicas como subir, bajar.

En cambio [24] Nishimura y Takashi han realizado un estudio en donde se analiza el comportamiento emergente. Ellos se enfocan en agrupamiento de individuos y describen varios comportamientos como estrategias colectivas. También se centran en el cómo dichas estrategias surgen de comportamientos individuales.

En su trabajo analizan la interacción que existe entre las presas y los depredadores utilizando un paradigma orientado a juegos para interpretar los resultados. La segunda consideración que ellos toman es que ambos tienen un grado de libertad muy amplio. Y el tercer factor que introducen en su modelo es el dinamismo de la evolución.

Tiene una gran aportación pues con teoría de juegos analiza los comportamientos emergentes, e inclusive agrega coevolución entre las presas y los depredadores. Ellos concluyen que las presas tienden a agruparse y descubrieron que la aleatoriedad genera simbiosis[24]. Este trabajo es una gran aportación a lo que es evolución, comportamiento emergente en los simuladores depredadores presa pero tiene la limitación de no integrar el aprendizaje como una herramienta extra en el desenvolvimiento de las presas.

Finalmente, nuestro trabajo es muy importante porque necesitamos un sistema que tenga aprendizaje y analice comportamientos emergentes para contribuir y ampliar en los estudios del efecto Baldwin y explorar que las técnicas computacionales desarrolladas son de gran utilidad.

3. TRABAJO PROPUESTO

3.1 Ambiente

En este proyecto de tesis se propone implementar un modelo computacional que simule un sistema depredador presa artificial. El modelo consiste en un ambiente toroidal de dos dimensiones delimitado por un rectángulo en el cual presas representadas por triángulos amarillos y el depredador está representado por triángulo verde (ver figura 3.1)

Se eligió una malla bidimensional debido que queremos observar reacciones emergentes y complejas en problemas sencillos. Se decidió un rectángulo toroidal porque se desea que nuestras presas tengan una ambiente dinámico y un ambiente sin obstáculos.

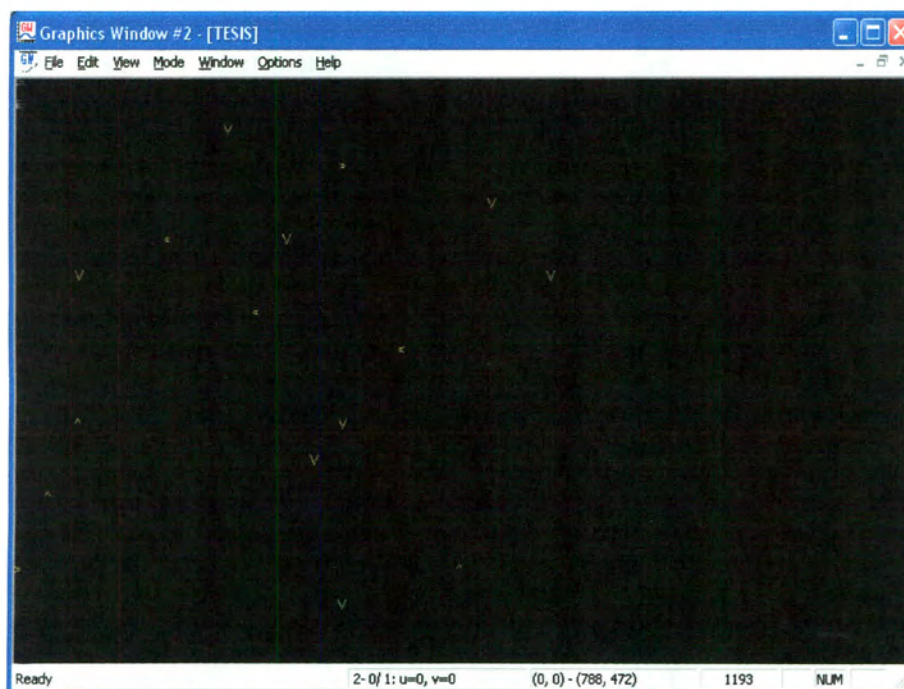


Fig. 3.1 Ambiente del sistema

El sistema está simulado por una malla toroidal en donde los agentes pueden desplazarse. Se asume que en el momento en que el depredador ocupe una celda cazará a todas las presas que encuentren en ésta como se muestra en la figura 3.2. [24]

La simulación se realiza en pasos discretos porque facilita la evaluación del tiempo de vida del sistema. Se mueven primero las presas, después se mueve el depredador, el depredador come las presas que caen en su celda y después se mueve y vuelve a comer las presas que están en la nueva celda.

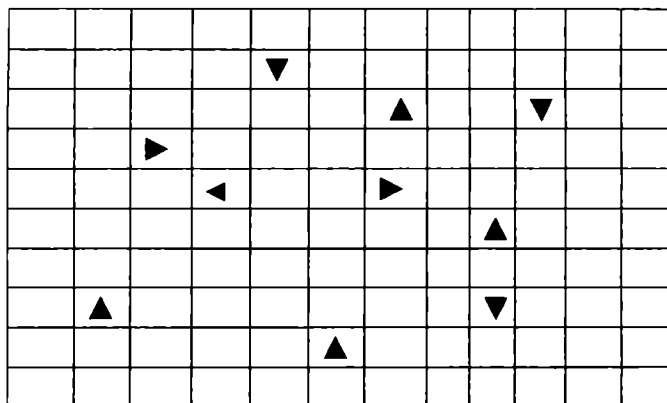


Fig. 3.2 Representación de la malla de movimiento

3.1.1 Presas

En nuestro ambiente existen dieciséis presas. Si las celdas que rodean a cada presa en nuestro modelo son ocho, consideramos duplicar ese número para que la presa pueda tener la opción de tener todas las celdas que le rodean ocupadas por otra presa o el depredador. Las presas tienen la capacidad de percibir las tres celdas que se ubican frente a ellas. Las siguientes diagonales serán con respecto a la dirección del agente. La celda izquierda es el primer cuadro izquierdo consecutivo en la diagonal izquierda, la celda centro es el primer cuadro consecutivo enfrente del agente y la celda derecha es el primer cuadro derecho consecutivo en la diagonal derecha (ver figuras 3.3, 3.4, 3.5, 3.6).



Fig. 3.3 Representación de la percepción de las presas al norte

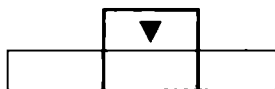


Fig. 3.4 Representación de la percepción de las presas al sur

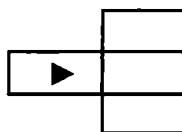


Fig. 3.5 Representación de la percepción de las presas al este

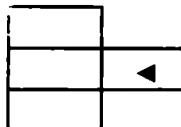


Fig. 3.6 Representación de la percepción de las presas al oeste

A continuación se muestran las posibles combinaciones de percepción de las presas:

PERCEPCIÓN	IZQUIERDA	CENTRO	DERECHA
CASO 1			
CASO 2			PRESA
CASO 3			DEPREDADOR
CASO 4		PRESA	
CASO 5		PRESA	PRESA
CASO 6		PRESA	DEPREDADOR
CASO 7		DEPREDADOR	
CASO 8		DEPREDADOR	PRESA
CASO 9	PRESA		
CASO 10	PRESA		PRESA
CASO 11	PRESA		DEPREDADOR
CASO 12	PRESA	PRESA	
CASO 13	PRESA	PRESA	PRESA
CASO 14	PRESA	PRESA	DEPREDADOR
CASO 15	PRESA	DEPREDADOR	
CASO 16	PRESA	DEPREDADOR	PRESA
CASO 17	DEPREDADOR		
CASO 18	DEPREDADOR		PRESA
CASO 19	DEPREDADOR	PRESA	
CASO 20	DEPREDADOR	PRESA	PRESA

Tabla 3.1 Percepciones de las presas

La tabla 3.1 muestra la percepciones de las presas, por ejemplo en el caso 8 la presa en su diagonal izquierda percibe la celda vacía, en la celda de enfrente percibe el depredador, en su diagonal derecha percibe una presa.

Las presas se pueden mover a las siguientes direcciones posibles: norte, sur, este, oeste, y las acciones de movimiento son: avanzar una celda en la dirección decidida o simplemente permanecer en la celda actual y girar en la dirección decidida como se muestra en la figura 3.7.

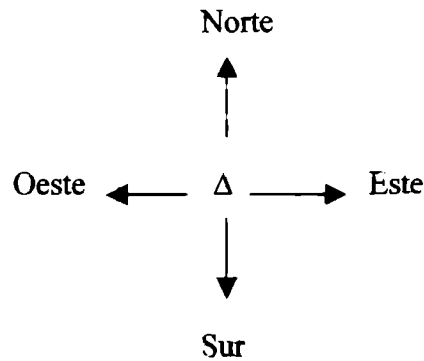


Fig. 3.7 Representación de los movimientos de las presas

Las presas con respecto a su dirección tienen las siguientes opciones:

1. Permanecer en la dirección actual
2. Cambio de dirección hacia su izquierda
3. Cambio de dirección hacia su derecha
4. Cambio de dirección hacia atrás

Las presas con respecto a su dirección tienen las siguientes opciones de movimiento:

1. Permanece en la dirección actual
2. Avanza una celda en la dirección actual.

Las combinaciones de movimiento de las presas son ocho y se muestran en a continuación:

MOVIMIENTO	IZQUIERDA	DERECHA	ADELANTE	ATRÁS
AVANZAR	X	X	X	X
PERMANECER	X	X	X	X

Tabla 3.2 Movimientos de las presas

La tabla 3.2 presentada muestra que es posible girar a la izquierda y avanzar, girar a la derecha y avanzar, permanecer en la dirección actual y avanzar, girar 180 grados y avanzar, o simplemente girar a la izquierda, girar a la derecha, mantener la dirección o girar 180 grados.

3.1.2 Depredador

En nuestro ambiente existe un depredador por que consideramos que es suficiente para representar el medio ambiente pues me he inspirado en la naturaleza [5]. Por su parte, el depredador tiene la capacidad de percibir las tres celdas que se ubican frente a ellas. Las siguientes aseveraciones serán con respecto a la dirección del agente. La celda izquierda es el primer cuadro izquierdo consecutivo en la diagonal izquierda, la celda centro es el primer cuadro consecutivo enfrente del agente y la celda derecha es el primer cuadro derecho consecutivo en la diagonal derecha (ver figuras 3.8, 3.9, 3.10, 3.11).



Fig. 3.8 Representación de la percepción del depredador al norte

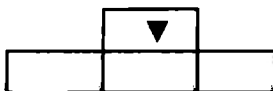


Fig. 3.9 Representación de la percepción del depredador al sur

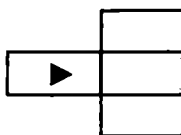


Fig. 3.10 Representación de la percepción del depredador al este

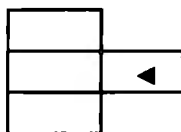


Fig. 3.11 Representación de la percepción del depredador al oeste

A continuación se muestran las posibles combinaciones de percepción del depredador:

PERCEPCIÓN	IZQUIERDA	CENTRO	DERECHA
CASO 1			
CASO 2			PRESA
CASO 3		PRESA	
CASO 4		PRESA	PRESA
CASO 5	PRESA		
CASO 6	PRESA		PRESA
CASO 7	PRESA	PRESA	
CASO 8	PRESA	PRESA	PRESA

Tabla 3.3 Percepciones de depredador

La tabla 3.3 muestra las percepciones del depredador, por ejemplo en el caso 3 el depredador en su diagonal izquierda percibe la celda vacía, en la celda de enfrente percibe una presa, en su diagonal derecha percibe una celda vacía.

El depredador se puede mover a las siguientes direcciones posibles: norte, sur, este, oeste, y las acciones de movimiento son: avanzar una celda a la dirección decidida como se muestra en la figura 3.12.

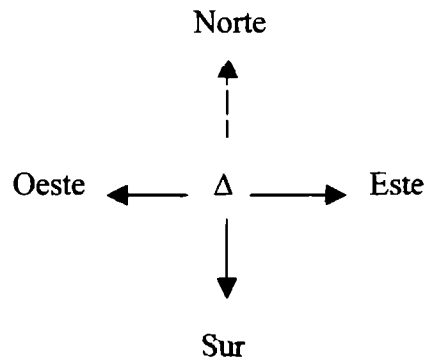


Fig. 3.12 Representación de los movimientos del depredador

El depredador con respecto a su dirección tiene las siguientes opciones:

1. Continuar en la misma dirección
2. Cambio de dirección hacia su izquierda
3. Cambio de dirección hacia su derecha
4. Cambio de dirección hacia atrás

El depredador con respecto a su dirección tiene la siguiente opción de movimiento:

1. Avanza una celda en la dirección actual.

Las combinaciones de movimiento de las presas son cuatro y se muestran en la siguiente tabla:

MOVIMIENTO	IZQUIERDA	DERECHA	ADELANTE	ATRÁS
AVANZAR	X	X	X	X

Tabla 3.4 Movimiento del depredador

La tabla 3.4 presentada a continuación muestra que para el depredador es posible girar a la izquierda y avanzar, girar a la derecha y avanzar, permanecer en la dirección actual y avanzar, girar 180 grados y avanzar.

3.2 Representación Interna

3.2.1 Presas

Las presas están representadas cada una por una red neuronal que permite determinar los movimientos de las presas a partir de sus percepciones en el ambiente. Se eligió una red neurona porque consideramos que una alternativa computacional válida para clasificar, decidir y producir una salida con base a parámetros de entrada.

También se escogió esta alternativa porque la idea ponderar las percepciones para producir salidas se considera más apegada a la naturaleza[5]. En la siguiente figura 3.13 se muestra la red que representa como sigue:

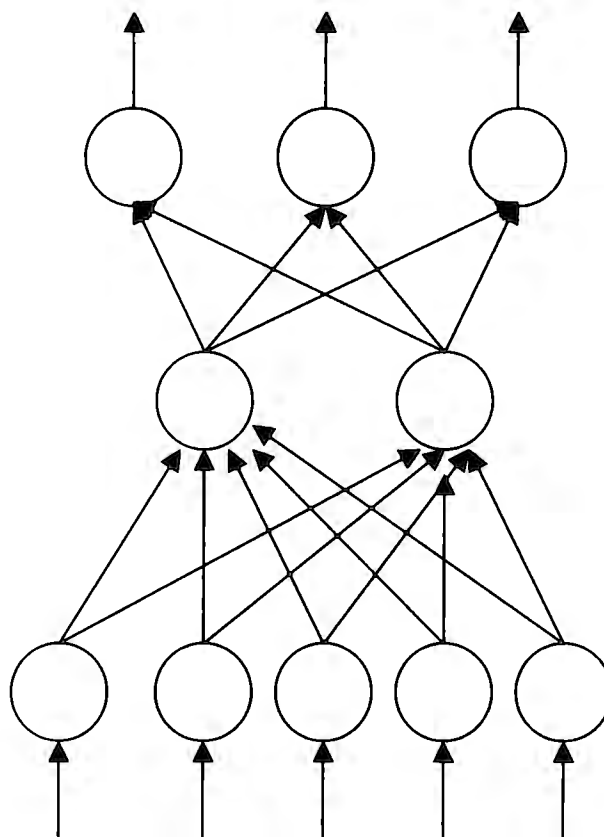


Fig. 3.13. Representación de la red neuronal de la presa

Como se menciona anteriormente, el espacio de las percepciones de las presas consiste en veinte observaciones posibles (Ver tabla 3.1.). Dado que las entradas a nuestro sistema son binarias, se requieren cinco unidades de procesamiento en la capa de entrada para que la presa perciba que está sucediendo en el mundo que le rodea.

Como puede observarse en la figura 3.13. existen dos neuronas en la capa intermedia para hacer más rápida la simulación. También las redes neuronales con menos unidades en la capa interna tiende a generalizar mejor, es decir, a producir mejores respuestas ante situaciones que nunca han enfrentado [25]. Finalmente existen tres neuronas en la capa de salida. Dado que el espacio de las acciones consiste en ocho movimientos posibles, se requieren tres unidades de procesamiento con valores binarios en la capa de salida.

3.2.1.1 Especificación de los pesos de conexión como cromosoma

La representación de la red neuronal como cromosoma requiere determinar el número de bits que necesitamos para representar los pesos de cada conexión. Las fórmulas que se utilizan fueron realizadas para este trabajo.

Supóngase que:

A el límite inferior o valor mínimo que puede tener un peso en decimal
B el límite superior o valor máximo que puede tener un peso en decimal
L la distancia euclidiana del límite inferior al límite superior
P la precisión o exactitud de los pesos
N el número entero de bits del gen que se necesitan para representar un peso
I el número de pesos de cada presa
C el número de bits del cromosoma que se necesita para representar una presa
K el número de presas en el ambiente
G el número de bits del genoma para representar el ambiente
Entero es una función que nos regresa la parte entera de un número
Round es una función que nos regresa el número redondeado, por ejemplo,
 si un número es igual a 0.5 la función round nos entregará 1

Entonces decimos que:

$$L = |B - A|$$

$$N = \text{entero}(\text{round}(0.499 + \log_2(10^P * L)))$$

$$C = \frac{N}{I}$$

$$G = C * K$$

Para convertir esa representación binaria asumimos que el valor de los pesos está entre menos diez y diez para los dieciséis pesos que existen en la representación de cada presa. Escogimos este rango porque queremos observar que tanto los pesos se aproximan al cero pues se considera que en la mayoría de las redes neuronales el aprendizaje se da cuando los pesos se encuentran cercanos a cero. También asumimos que queremos una precisión de cuatro decimales pues aunque este es un costo computacional muy grande pues el tamaño del genoma y el procesamiento aumenta es mejor tener genomas muy precisos para garantizar un buen aprendizaje. Dadas estas especificaciones obtenemos que:

$$L = |10 - (-10)| = 20$$

$$N = \text{entero}(\text{round}(0.499 + \log_2(10^4 * 20)))$$

$$N = 18$$

Entonces el número de bits que necesitamos para representar una conexión o peso de la presa es 18 y se expresa en la siguiente figura:

1	1	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0
17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Fig. 3.14 Representación interna del peso de conexión

Aquí se expresa el tamaño de peso indicando los alelos que se van acomodando de tal forma que llenan los 18 espacios que se necesitan para representar un peso.

Dado que la presa requiere 16 conexiones o pesos para percibir el ambiente que hemos diseñado tenemos que el número de bits (C) que necesita la presa para representar los pesos que a su vez contienen información del ambiente es:

$$C = 18 * 16 = 288 \text{ bits}$$

Es decir, el número C es aquel que representa los genes que se necesitan para codificar el cromosoma o número de bits que se necesitan para representar una presa como se expresa en la siguiente figura:

1	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	0	0	
287																	17						0

Fig. 3.15 Representación interna de los pesos de conexión de una presa

El número de bits que se necesitan para representar una población de 1000 individuos resulta de multiplicar el número de bits que requiere una individuo por el número de individuos en la población que compone la generación.

$$G = 288 * 1000 = 288\ 000 \text{ bits}$$

Tendremos entonces una población de 1000 individuos representados en grupos de N bits que representan los pesos de todas las presas como se muestra en la siguiente figura:

	1	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	0	0
287999																		17					0

Fig. 3.16 Representación de la generación con los 1000 individuos

3.2.1.2 Codificación decimal de los pesos.

Supóngase que:

A el límite inferior

B el límite superior

L la distancia euclidiana del límite inferior al límite superior

N el número entero de bits que se necesitan para representar un peso

M el resultado de la función que transforma un número N de bits de binario a decimal.

Ahora se convierte la representación del peso en binario a decimal de tal forma que resulte el peso con la precisión P dentro del rango $[A,B]$. Para esto se utiliza la siguientes fórmula:

$$\text{Peso} = A + M * L / 2^N$$

Esta fórmula se aplica para cada peso o conjunto de N bits de cada individuo para todos los individuos del algoritmo genético. Por ejemplo:

Supóngase que el primer gen dentro del primer cromosoma es el que está en la siguiente figura:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Fig. 3.17 Representación del primer gen dentro del primer cromosoma

Entonces el valor de $M = 5$ y aplicando la fórmula para obtener el peso dentro del rango y con la precisión deseada tenemos que el peso = $-10 + 5 * 20 / 2^{18} = -9.9996$.

3.2.2 Depredador

El depredador estará descrito por un algoritmo computacional o código que selecciona norte, sur, este u oeste. Se van a ver que ocurre cuando el ambiente es muy dinámico, constante y cuando no lo es tanto. Se decidió utilizar varias alternativas para explorar la viabilidad de los fenómenos de adaptación bajo diferentes circunstancias. Los algoritmos se expresan a continuación:

```

Si contador constante == Numero celdas para ser aleatorio
  Selecciona aleatoriamente una dirección ( norte, este, sur, oeste).
  Constante es igual a cero
Si no
  Selecciona la dirección constante y suma contador constante
  Si existe una presa en el campo de visión
    Moverse a la dirección un espacio ( Ej. norte)
  Si no existe una presa en el campo de visión
    Girar a la dirección consecutiva a las manecillas del reloj
    Si existe una presa en el campo de visión
      Moverse a la dirección un espacio (Ej. Este)
    Si no existe presa en el campo de visión
      Girar a la dirección consecutiva a las manecillas del reloj
      Si existe una presa en el campo de visión
        Moverse a la dirección un espacio ( Ej. Sur)
      Si no existe presa en el campo de visión
        Girar a la dirección consecutiva a las manecillas del reloj
        Si existe una presa en el campo de visión
          Moverse a la dirección un espacio ( Ej. Oeste)
        Si no existe presa en el campo de visión
          Moverse a la dirección aleatoria original ( Ej. Norte)

```

Tabla 3.5 Pseudocódigo del depredador semi aleatorio

```

Selecciona la dirección constante-
  Si existe una presa en el campo de visión
    Moverse a la dirección un espacio ( Ej. norte)
  Si no existe una presa en el campo de visión
    Girar a la dirección consecutiva a las manecillas del reloj
    Si existe una presa en el campo de visión
      Moverse a la dirección un espacio (Ej. Este)
    Si no existe presa en el campo de visión
      Girar a la dirección consecutiva a las manecillas del reloj
      Si existe una presa en el campo de visión
        Moverse a la dirección un espacio ( Ej. Sur)
      Si no existe presa en el campo de visión
        Girar a la dirección consecutiva a las manecillas del reloj
        Si existe una presa en el campo de visión
          Moverse a la dirección un espacio ( Ej. Oeste)
        Si no existe presa en el campo de visión
          Selecciona la dirección constante ( Ej. Norte)

```

Tabla 3.6 Pseudocódigo del depredador constante

```

Selecciona la dirección constante
  Si existe una presa en el campo de visión
    Moverse a la dirección un espacio ( Ej. norte)
  Si no existe una presa en el campo de visión
    Girar a la dirección consecutiva a las manecillas del reloj
    Si existe una presa en el campo de visión
      Moverse a la dirección un espacio (Ej. Este)
    Si no existe presa en el campo de visión
      Girar a la dirección consecutiva a las manecillas del reloj
      Si existe una presa en el campo de visión
        Moverse a la dirección un espacio ( Ej. Sur)
      Si no existe presa en el campo de visión
        Girar a la dirección consecutiva a las manecillas del reloj
        Si existe una presa en el campo de visión
          Moverse a la dirección un espacio ( Ej. Oeste)
        Si no existe presa en el campo de visión
          Selecciona la dirección constante ( Ej. Norte)

```

Tabla 3.7 Pseudocódigo del depredador constante

3.3 Fases del trabajo propuesto

La idea de este trabajo es que contribuya al entendimiento del proceso evolutivo, el proceso de aprendizaje y la interacción entre los mismos. Así también, contribuir a la comprensión del comportamiento emergente en un modelo computacional depredador presa, al diseño de algoritmos que puedan ayudar a resolver problemas computacionales complejos y finalmente, el resultado esperado es comprobar que el aprendizaje guía la evolución en este modelo.

En nuestro modelo se desarrolla un algoritmo para poder simular la evolución y otro algoritmo para simular la evolución con aprendizaje. Se realizarán dos experimentos. En el primer experimento se manejan los factores que influyen en la evolución, y en el segundo experimento se manejan los factores que influyen en la evolución con aprendizaje en un ambiente determinado. En el primero se incluyen etapas como la selección, fase de cruzamiento, probabilidades de mutación como parte del proceso evolutivo de los individuos y en el segundo

se incluyen etapas como la selección con aprendizaje, fase de cruzamiento, probabilidades de mutación como parte del proceso evolutivo de los individuos.

Finalmente habrá una fase de observación en donde analizaremos gráficamente ya sea la etapa de evolución y la etapa de evolución con aprendizaje. La fase de observación será una interfaz gráfica en donde se toma como entrada los pesos entrenados por alguno de los dos procesos: evolución o evolución con aprendizaje (“efecto Baldwin”).

3.3.1 Evolución

Como puede observarse en la figura. 3.13 la red neuronal consta tres capas. Existen cinco neuronas en la capa de entrada, dos neuronas en la capa intermedia y tres neuronas en la capa de salida. También podemos ver que hay líneas que conectan a dichas neuronas. Para esta red neuronal existen diez conexiones entre la capa de entrada y la capa intermedia. Asimismo, existen seis conexiones entre la capa intermedia y la capa de salida.

Como resultado del proceso evolutivo los pesos cambiarán paulatinamente y por ende el comportamiento de las presas. Llamaremos gen al conjunto de bits que representan un peso, llamaremos alelo al valor 0 o 1 de cada bit dentro del gen, llamaremos cromosoma o genoma al conjunto de bits que se necesitan para representar los pesos de una presa.

3.3.1.1 Proceso de Evolución

Para evolucionar tomaremos los cromosomas en binario y empezaremos el proceso de Evolución. El proceso de evolución consiste en la siguiente tabla:

```

Inicializar los cromosomas aleatoriamente
Mientras no se llegue a la última generación o al más óptimo
  Selección de los cromosomas más aptos según la función de aptitud
  Cruzamiento
  Mutación
  Guardar los pesos resultado de la generación
Fin del mientras

```

Tabla 3.8 Pseudocódigo del proceso de evolución

3.3.1.1.1 Inicializar los cromosomas aleatoriamente

Dado que tenemos una población de mil individuos iniciamos aleatoriamente los mil cromosomas con valores de 0 o 1. Utilizamos mil individuos porque las reglas empíricas indican que una población que cuadriplica la longitud del genoma da buenos resultados.

3.3.1.1.2 Selección de los cromosomas mas aptos según la función de aptitud

Para simular la selección se utiliza una función llamada la función de aptitud. El algoritmo de selección primero calcula la aptitud para cada cromosoma y regresa una calificación al cromosoma basada el tiempo de vida del sistema con 16 presas que tienen el mismo cromosoma. El algoritmo se muestra en la siguiente tabla siguiente:

```

Se cambia el cromosoma de 288 bits mediante el proceso de transformación de
binario a decimal
Se elige un numero 16 de presas.
Se inicializa la posición de las presas y el depredador.
Inicializa el tiempo de vida del sistema en cero
Mientras sigan vivas todas las presas y no acabe el tiempo de vida del ambiente
  Iniciar el numero de presas muertas en cero
  Para todas las presas
    Si está viva la presa
      Inserta percepción del ambiente a la presa
      Realiza movimiento de la presa
  Fin para todas las presas.
  Mata las presas en la vieja posición del depredador
  Realiza movimiento del depredador
  Mata las presas en la nueva posición del depredador
  Para todas la presas
    Si está viva la presa
      Incrementa el tiempo de vida del sistema
    Si está muerta la presa
      Incrementa el número de presas muertas
  Fin para todas las presas
Fin del mientras
Regresar valor del tiempo de vida de la presa

```

Tabla 3.9 Pseudocódigo de la función de aptitud de la evolución

Se eligió realizar el método de selección proporcional a la aptitud porque como la aptitud está basada en el tiempo de vida de las presas se nos hizo más apegado a la naturaleza utilizar este método.

Es muy importante observar que este algoritmo se realiza para cada cromosoma de la generación. Se escogió la suma de las presas como el tiempo de vida debido a que experimentos previos han proporcionado buenos resultados [4]. Es una forma de separar aprendizaje y evolución en donde ambos procesos persiguen un objetivo común que hacer que las presas vivan más.

3.3.1.1.3 Cruzamiento

Después de evaluar los cromosomas se seleccionan los más aptos según su calificación resultante de la función de aptitud y se lleva a cabo el cruzamiento como se muestra en la siguiente figura:

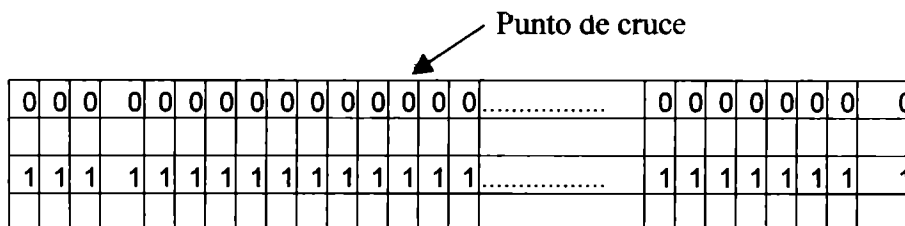


Fig. 3.18 Representación de los cromosomas antes del cruzamiento

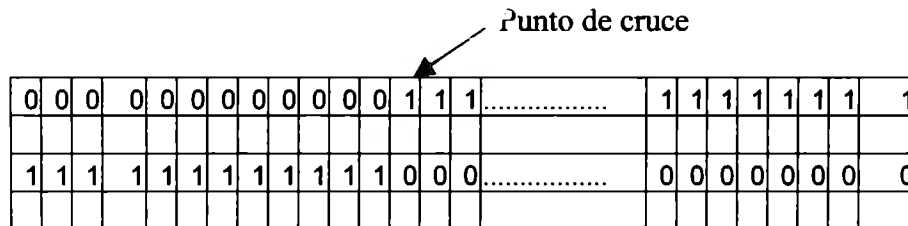


Fig. 3.19 Representación de los cromosomas después del cruzamiento.

Como se puede observar en la figura 3.18 se tienen 2 cromosomas. Se forman parejas y según el porcentaje de cruzamiento se elige un número donde se lleva a cabo el corte. Después el conjunto de dígitos se intercambia y se producen dos descendientes muestra en la figura 3.19. Se utilizó un punto de cruce 0.6. [17]

3.3.1.1.4 Mutación

La mutación es un proceso en el que se intercambia el alelo de la posición seleccionada, es decir, según el porcentaje de mutación cambia el bit de la posición. Si existe un 0 en el bit seleccionado intercambia por 1 y si existe un 1 se intercambia por un 0 simulando el efecto de mutación. Se utilizó un punto de cruce de 0.001. [17]

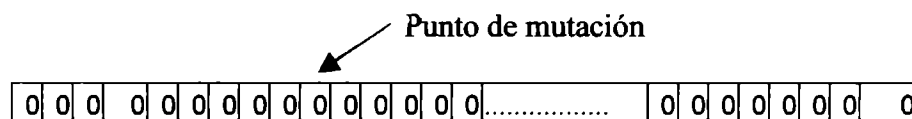


Fig. 3.20 representación de los cromosomas antes de mutación

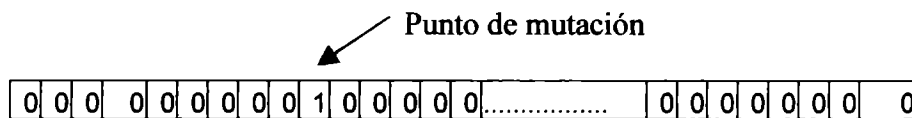


Fig. 3.21 Representación de los cromosomas después de la mutación.

3.3.1.1.5 Guardar los pesos resultado de la generación

Cada generación después de haber llevado a cabo la evolución se guardarán los pesos que resultan de la misma en un archivo de tal forma que el módulo que gráfica el resultado pueda tomar los pesos y colocárselos a las presas que se van a observar en el simulador.

3.3.2 Evolución con aprendizaje (efecto Baldwin)

Como puede observarse en la figura. 3.13 la red neuronal consta tres capas. Existen cinco neuronas en la capa de entrada, dos neuronas en la capa intermedia y tres neuronas en la capa de salida. También podemos ver que hay líneas que conectan a dichas neuronas. Para esta red neuronal existen diez conexiones entre la capa de entrada y la capa intermedia. Asimismo, existen seis conexiones entre la capa intermedia y la capa de salida.

Al igual que en el caso del proceso evolutivo, los pesos cambiarán gradualmente y por ende el comportamiento de las presas. Aquí también llamaremos gen al conjunto de bits que representan un peso, llamaremos alelo al valor 0 o 1 de cada bit dentro del gen, llamaremos cromosoma o genoma al conjunto de bits que se necesitan para representar los pesos de una presa.

Se eligió el algoritmo de aprendizaje que refuerza positivamente aquellas acciones que hacen que continúe viva la presa y refuerza negativamente aquellas acciones que hacen que se muera una presa. Este algoritmo de aprendizaje es no supervisado y se escogió porque está apegado a la naturaleza y a la forma instintiva de la misma. También se eligió porque es sencillo y rápido. Aún así, consideramos que más sofisticado que aprendizaje aleatorio empleado por Hinton y Nowlan [10]. Se decidió escoger una constante de aprendizaje muy pequeña de .0001 por la misma razón de que queremos que el comportamiento del sistema sea gradual y sea posible observar cambios en el tiempo de vida que sean susceptibles de interpretar.

3.3.2.1 Proceso de Evolución con aprendizaje

Para evolucionar tomaremos los cromosomas en binario y empezaremos el proceso de Evolución. El proceso de evolución consiste en la siguiente tabla:

```

Inicializar los cromosomas aleatoriamente
Mientras no se llegue a la última generación o al más optimo
  Selección de los cromosomas más aptos según la función de aptitud y
  aprendizaje.
  Cruzamiento
  Mutación
  Guardar los pesos resultado de la generación
Fin del mientras

```

Tabla 3.10 Pseudocódigo del proceso de evolución con aprendizaje

3.3.2.1.1 Inicializar los cromosomas aleatoriamente

Dado que tenemos una población de mil individuos iniciamos aleatoriamente los mil cromosomas con valores de 0 o 1.

3.3.2.1.2 Selección de los cromosomas mas aptos según la función de aptitud con aprendizaje

Para simular la selección se utiliza una función llamada la función de aptitud que agrega aprendizaje. El algoritmo de selección primero calcula la aptitud para cada cromosoma y regresa una calificación al cromosoma basada el tiempo de vida del sistema con 16 presas que tienen el mismo cromosoma. El algoritmo se muestra en la siguiente tabla siguiente:


```

Se cambia el cromosoma de 288 bits mediante el proceso de transformación de
binario a decimal
Se elige un numero 16 de presas.
Todas las presas utilizarán el cromosoma de 288 bits transformado a decimal
Se inicializa la posición de las presas y el depredador.
Inicializa el tiempo de vida del sistema en cero
Mientras sigan vivas todas las presas y no acabe el tiempo de vida del ambiente
  Iniciar el número de presas muertas en cero
  Se inicializa el delta aprendizaje en cero
  Para todas las presas
    Si está viva la presa
      Inserta percepción del ambiente a la presa
      Realiza movimiento de la presa
  Fin para todas las presas.
  Mata las presas en la vieja posición del depredador
  Realiza movimiento del depredador
  Mata las presas en la nueva posición del depredador
  Para todas la presas
    Si está viva la presa
      Incrementa el tiempo de vida del sistema
      Incrementa el delta aprendizaje
    Si está muerta la presa
      Incrementa el número de presas muertas
      Decrementa el delta aprendizaje
  Fin para todas las presas
  Suma el Delta Aprendizaje multiplicado por la constante de aprendizaje
  : Todos los pesos generados por el cromosoma que se esta evaluando.
  Modifica los pesos que genera el cromosoma más no el cromosoma.
Fin del mientras
Regresar valor del tiempo de vida de la presa

```

Tabla 3.11 Pseudocódigo de la función de aptitud de evolución con aprendizaje

3.3.2.1.3 Cruzamiento

Al igual que en el proceso de evolución pura el cruzamiento se muestra en la siguiente figura:

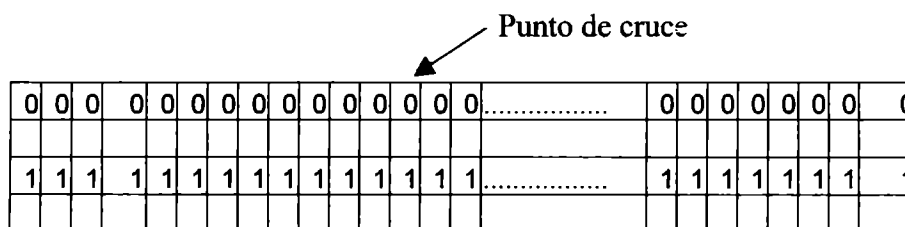


Fig. 3.22 Representación de los cromosomas antes del cruzamiento

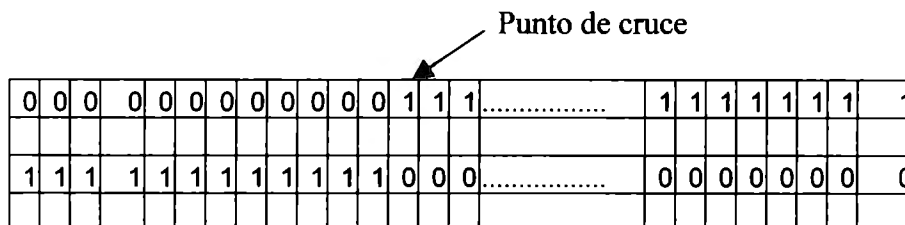


Fig. 3.23 Representación de los cromosomas después del cruzamiento.

Como se puede observar en la figura 3.22 se tienen 2 cromosomas. Se forman parejas y según el porcentaje de cruzamiento se elige un número donde se lleva a cabo el corte. Después el

conjunto de dígitos se intercambia y se producen dos descendientes muestra en la figura 3.23. También aquí se utilizó un punto de cruce 0.6. [17]

3.3.2.1.4 Mutación

En esta fase se intercambia el alelo de la posición seleccionada, es decir, según el porcentaje de mutación cambia el bit de la posición. Si existe un 0 en el bit seleccionado intercambia por 1 y si existe un 1 se intercambia por un 0 simulando el efecto de mutación. Se utilizó un punto de mutación de 0.001. [17]

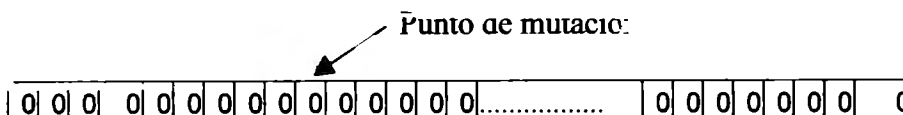


Fig. 3.24 Representación de los cromosomas antes de mutación

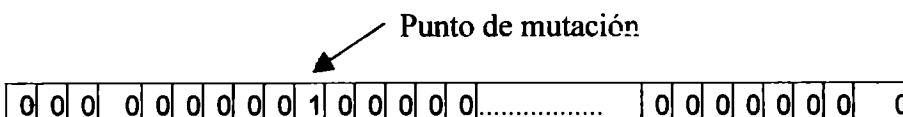


Fig. 3.25 Representación de los cromosomas después de la mutación.

3.3.2.1.5 Guardar los pesos resultado de la generación

Cada generación después de haber llevado a cabo la evolución se guardarán los pesos que resultan de la misma en un archivo de tal forma que el módulo que gráfica el resultado pueda tomar los pesos y colocárselos a las presas que se van a observar en el simulador.

3.3.3 Simulación

El simulador es un programa que con base a los pesos leídos de un archivo muestra el comportamiento de las presas y el depredador.

Como ya habíamos mencionado existen 2 procesos de entrenamiento para los pesos que toma de un archivo el simulador como parámetro. Estos procesos de entrenamiento son: evolución y efecto Baldwin o evolución con aprendizaje.

El simulador toma los pesos como parámetro y empieza a simular el ambiente, es decir, es un sistema en donde las entradas son la percepción del ambiente y las salidas las acciones de las presas y el depredador en donde se van graficando secuencialmente las acciones en donde primero se mueven las presas y al final el depredador. Simulando que cuando el depredador se encuentra en una posición donde se encuentre una presa ésta será removida del sistema.

4 EXPERIMENTOS Y RESULTADOS

4.1 Evolución

4.1.1 Experimentos y resultados

Los experimentos en la fase de evolución consistieron en correr el algoritmo genético y observar cómo cambian los tiempos de vida de las presas así como las simulaciones con los cromosomas más aptos. Se tomaron los pesos del cromosoma más apto obtenido de la evolución. Se copiaron al archivo de entrada del simulador y se observaron los comportamientos que tienen las presas en el simulador con los pesos evolucionados. En todos los experimentos usamos mil generaciones

Inicialmente, se realizaron pruebas colocando las presas y el depredador en posiciones al azar. Utilizamos 100 individuos por generación, un depredador aleatorio y obtuvimos lo siguiente:

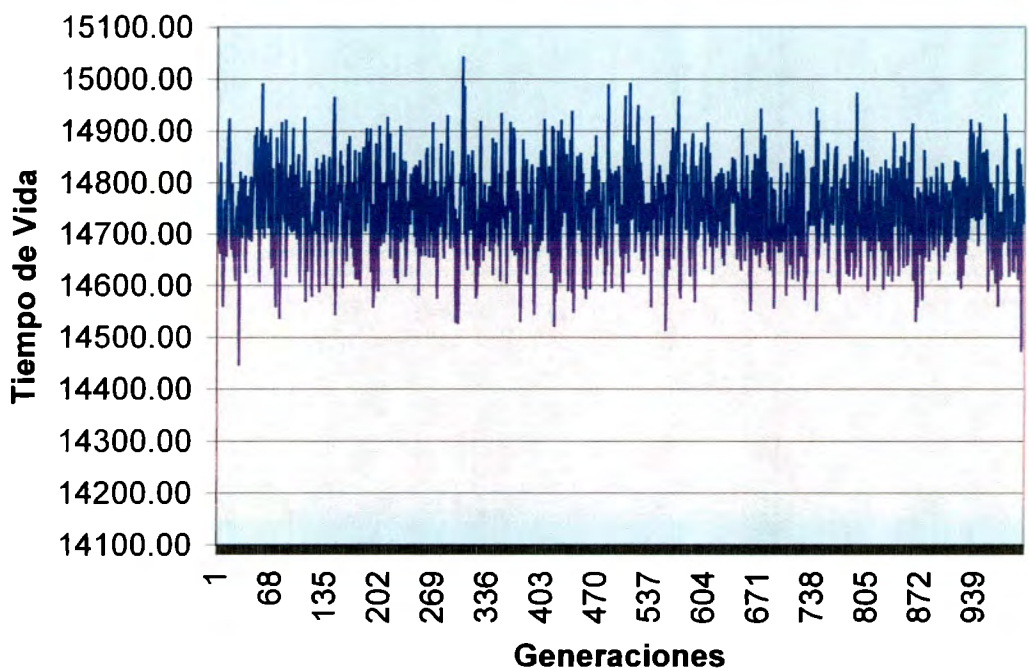


Fig. 4.1 Evolución 1000 generaciones 100 individuos con depredador aleatorio.

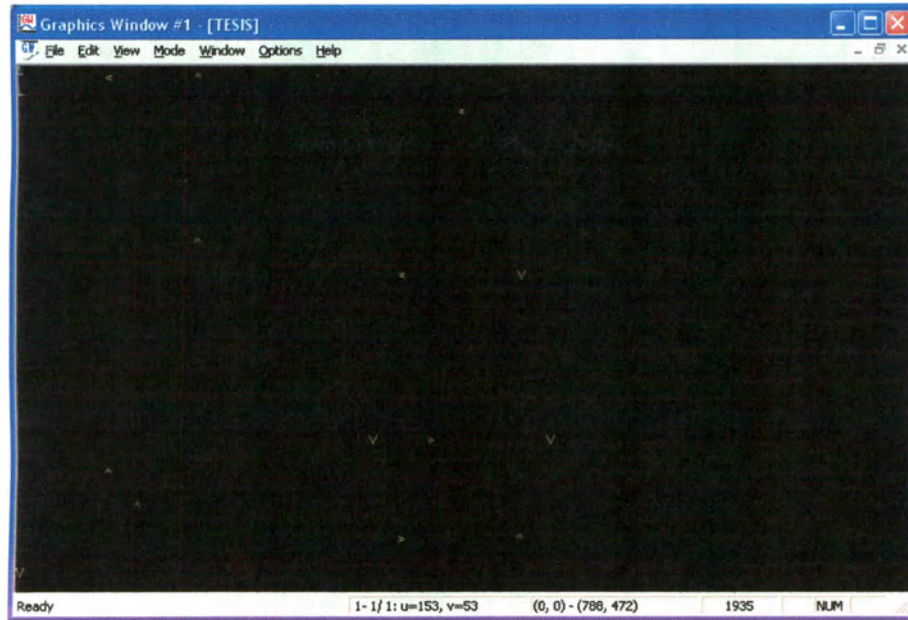


Fig. 4.2 Simulación 1000 generaciones 100 individuos con depredador aleatorio

Como se puede observar en la figura 4.1 y 4.2 el tiempo de vida inicia aprox. en 14700, termina 14500 y existen trece presas vivas. Después se cambio el sistema para que las presas empezaran ordenadas en la fila superior y el depredador en el centro del rectángulo. Se realizaron pruebas con depredador constante ahora con 1000 individuos por generación. Se obtuvo lo siguiente:

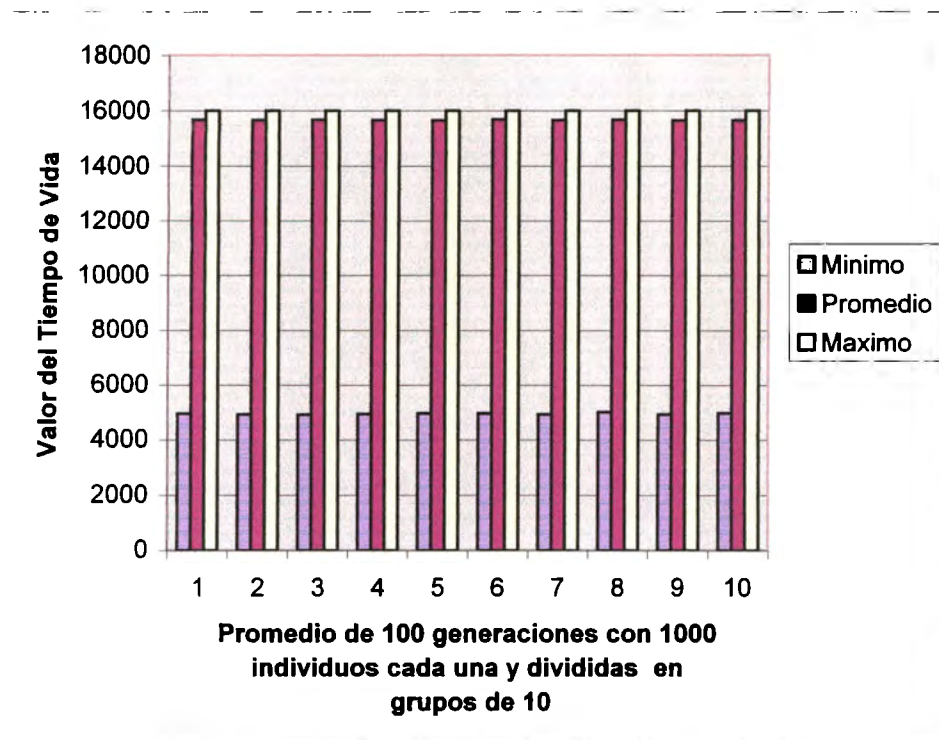


Fig. 4.3 Evolución 1000 generaciones 1000 individuos con depredador constante



Fig. 4.4 Simulación 1000 generaciones 1000 individuos con depredador constante

Se puede observar en la fig. 4.3 que el tiempo de vida de las presas incrementa hasta que se mantiene constante. En la fig. 4.4 se ve que el depredador se come solamente a una presa y las demás quedan vivas. Después cambiamos corrimos el sistema varias veces pero ahora con un depredador semi aleatorio.

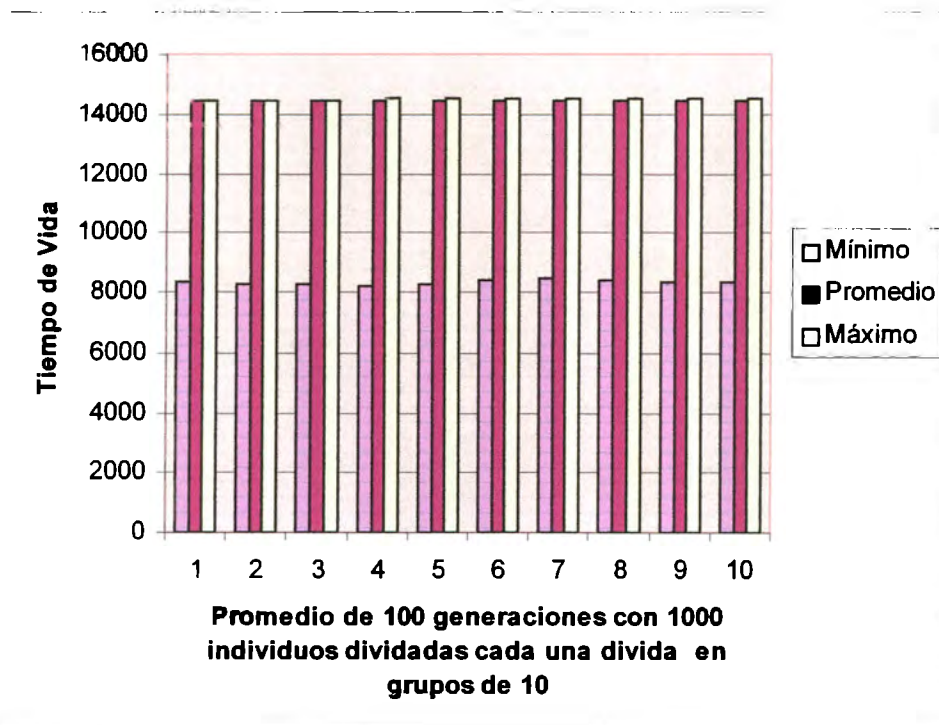


Fig. 4.5 Evolución 1000 generaciones 1000 individuos con depredador semi aleatorio



Fig. 4.6 Simulación 1000 generaciones 1000 individuos con depredador semi aleatorio

En la fig. 4.5 se observa que los tres tiempos de vida de vida, el máximo, el promedio y el mínimo va incrementando. Y en la figura 4.6 se observa que se generan grupos. Después decidimos reducir la malla toroidal para ejercer más presión sobre las presas y volvimos a probar con un depredador aleatorio y obtuvimos lo siguiente:

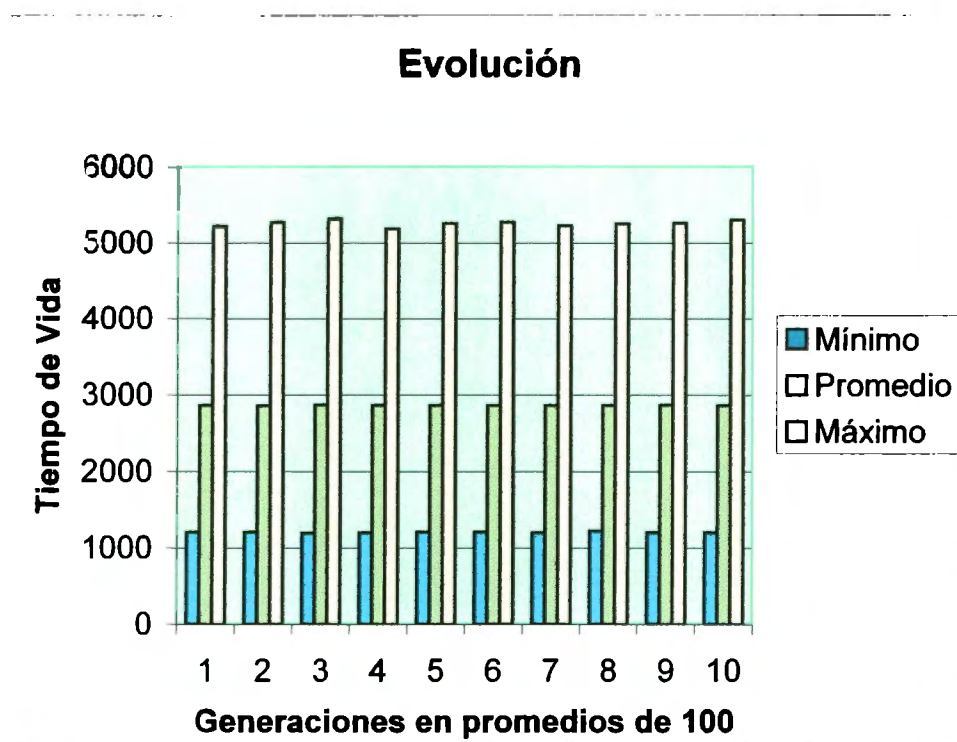


Fig. 4.7 Evolución 1000 generaciones 1000 individuos con depredador aleatorio v reducción de malla toroidal.

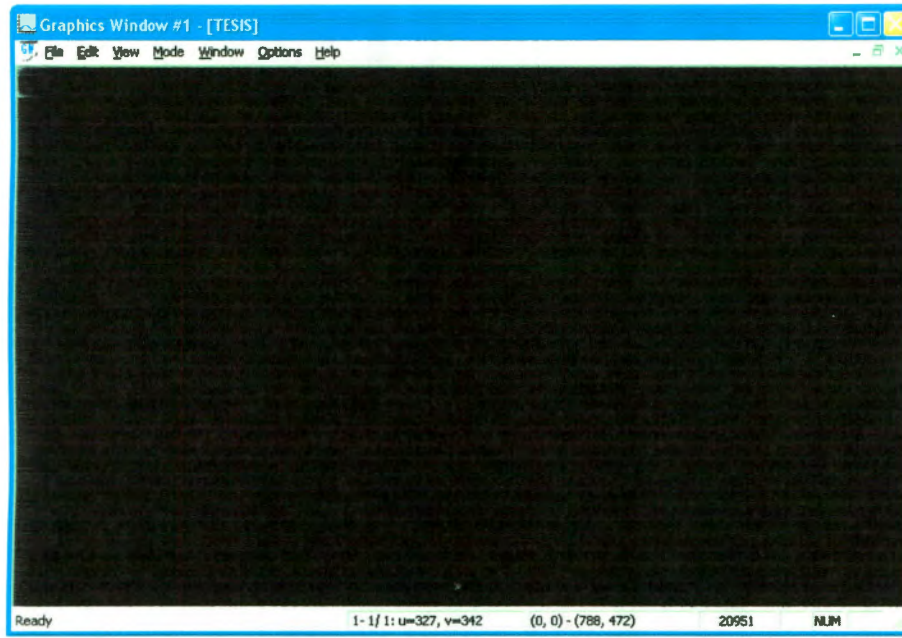


Fig. 4.8 Simulación 1000 generaciones 1000 individuos con depredador aleatorio y reducción de malla toroidal.

Como se puede observar en la figura 4.7 el tiempo de vida máximo esta oscilando y en la fig. 4.8 el depredador se come a todas las presas. Ahora continuamos las pruebas cambiando el depredador aleatorio por uno constante manteniendo reducida la malla toroidal y obtuvimos lo siguiente:

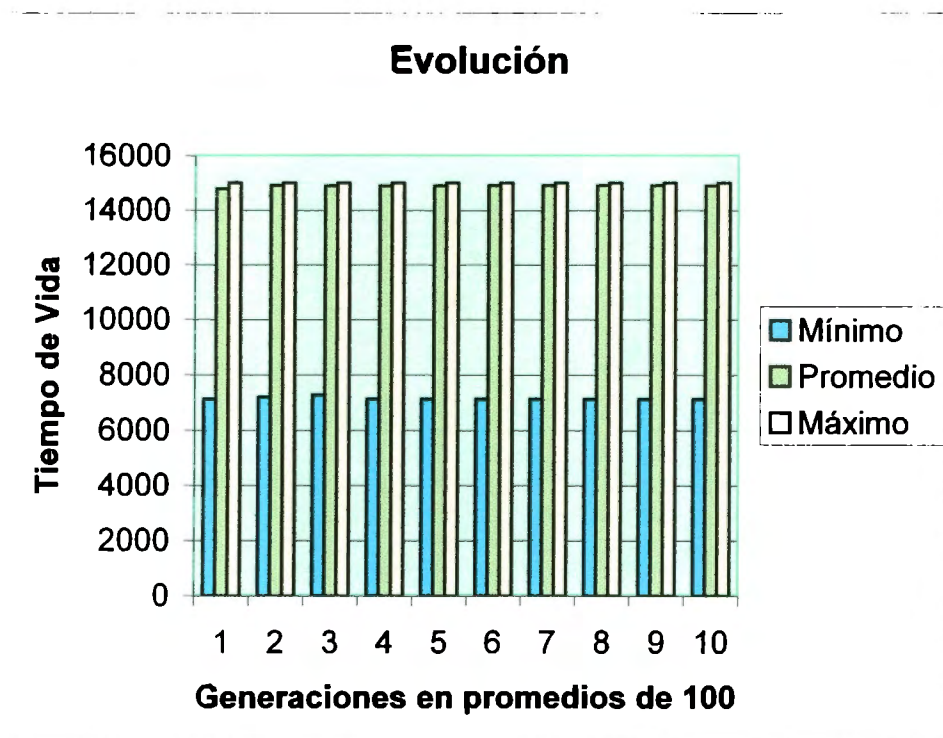


Fig. 4.9 Evolución 1000 generaciones 1000 individuos con depredador constante y reducción de malla toroidal.



Fig. 4.10 Simulación 1000 generaciones 1000 individuos con depredador constante y reducción de malla toroidal.

Como se puede observar en la fig. 4.9 el depredador solamente se come una presa y el tiempo de vida va incrementando hasta que se estabiliza y mantiene constante. En la figura 4.10 se observa que una presa se muere pero otra presa imita el movimiento del depredador. Finalmente utilizando el mismo depredador semi aleatorio decidimos reducir la malla toroidal, ejercer más presión y obtuvimos lo siguiente:

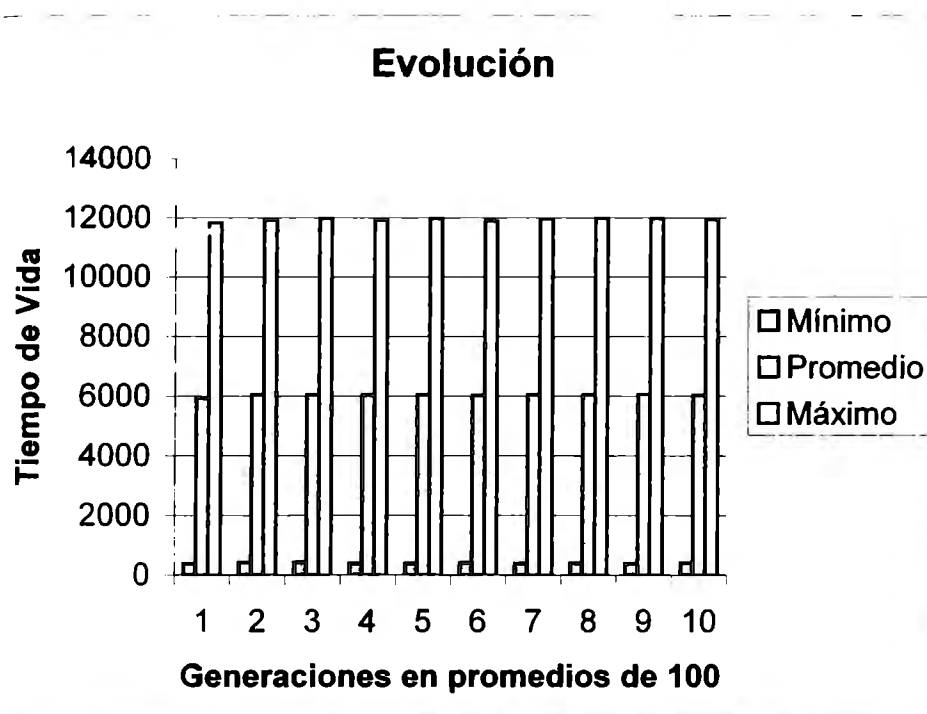


Fig. 4.11 Evolución 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de malla toroidal.



Fig. 4.12 Simulación 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de malla toroidal.

Como podemos observar en la figura 4.13 que el depredador mata solamente a trece presas y el tiempo de vida máximo, la presa más inteligente, oscila hacia arriba. El valor del tiempo de vida mínimo, la presa menos inteligente, aunque es muy pequeño oscila hacia arriba también. La presa mata a 13 presas.

4.2 Evolución con aprendizaje (efecto Baldwin).

4.2.1 Experimentos y resultados

En la fase de experimentos de evolución con aprendizaje se realizaron a las mismas pruebas que solamente utilizando evolución. En principio, se realizaron pruebas colocando las presas y el depredador en posiciones al azar. Utilizamos 100 individuos por generación, un depredador aleatorio y obtuvimos lo siguiente:

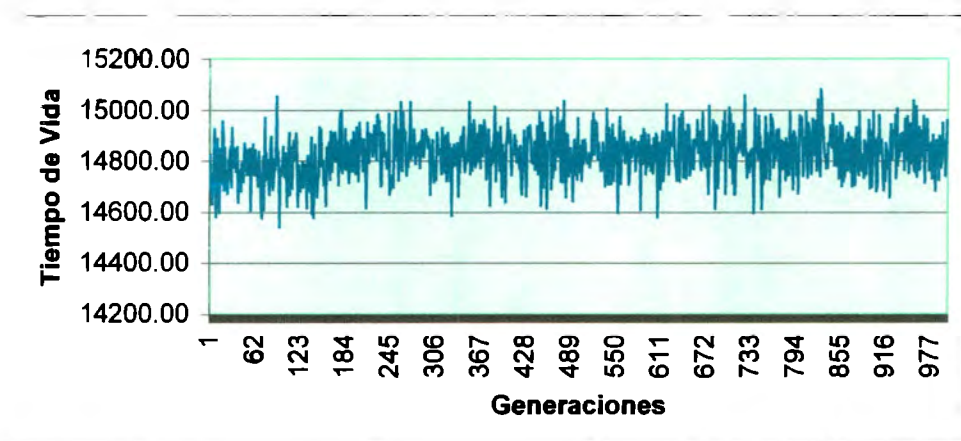


Fig. 4.13 Efecto Baldwin 1000 generaciones con 100 individuos aleatorio



Fig. 4.14 Simulación 1000 generaciones con 100 individuos con depredador aleatorio

Como podemos observar en la figura 4.13 la gráfica es oscilando en niveles de tiempo de vida con una ligera tendencia hacia arriba empieza con los 14,600 y acaba alrededor de los 15000. En la gráfica 4.14 existen 14 presas vivas. Después se cambio el sistema para que las presas empezaran ordenadas en la fila superior y el depredador en el centro del rectángulo. Se realizaron pruebas con depredador constante ahora con 1000 individuos por generación. Se obtuvo lo siguiente:

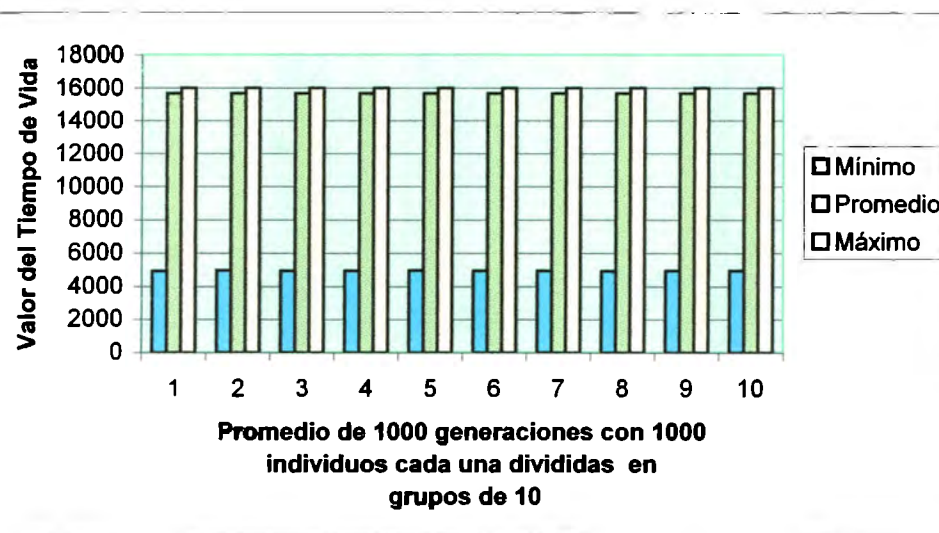


Fig. 4.15 Efecto Baldwin 1000 generaciones 1000 individuos con depredador constante



Fig. 4.16 Simulación 1000 generaciones con 1000 individuos con depredador constante

En la figura 4.15 observamos que el tiempo de vida máximo permanece constante en el óptimo. Y en la figura 4.16 el depredador solamente mata a una presas. Después cambiamos corrimos el sistema varias veces pero ahora con un depredador semi aleatorio. Obtuvimos lo siguiente:

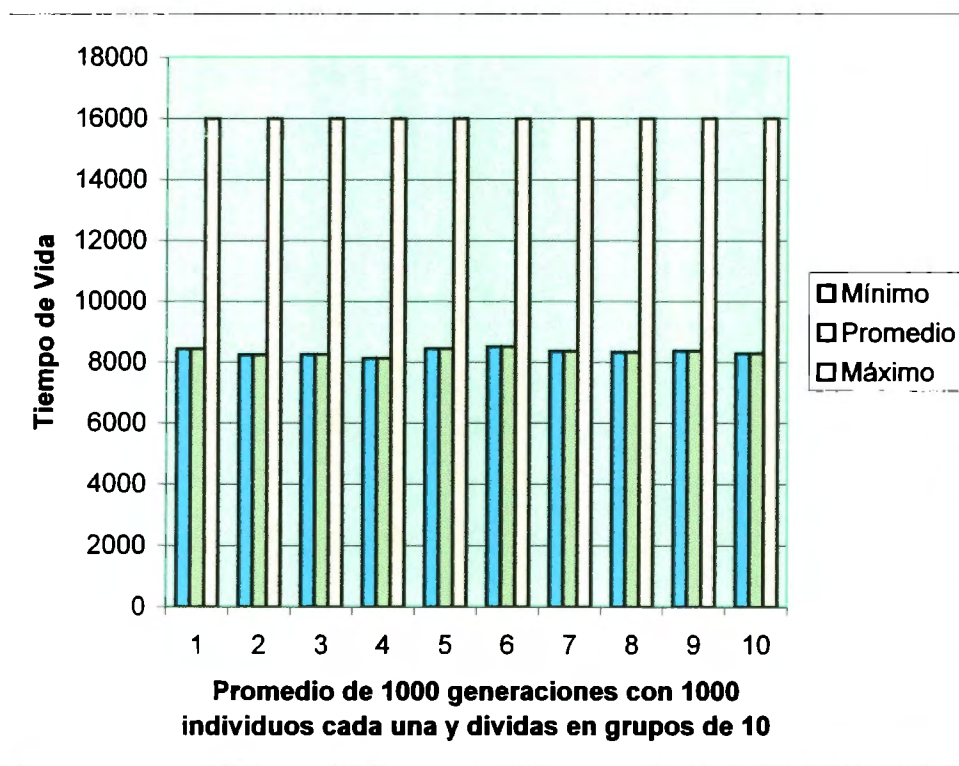


Fig. 4.17 Efecto Baldwin 1000 generaciones con 1000 individuos semi aleatorio



Fig. 4.18 Simulación 1000 generaciones con 100 individuos con depredador semi aleatorio

Como puede observarse en la figura 4.17 el sistema siempre hay uno muy bueno y el promedio va oscilando. En la gráfica 4.18 se observa una presa se separa y también se forman grupos. Después decidimos reducir la malla toroidal para ejercer más presión sobre las presas y volvimos a probar con un depredador aleatorio y obtuvimos lo siguiente:

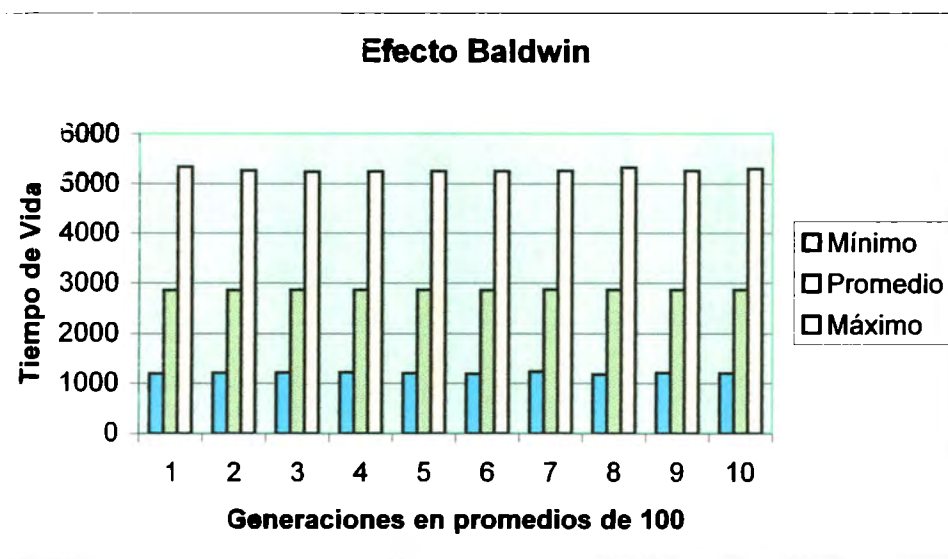


Fig. 4.19 Efecto Baldwin 1000 generaciones 1000 individuos con depredador aleatorio y reducción de malla toroidal.

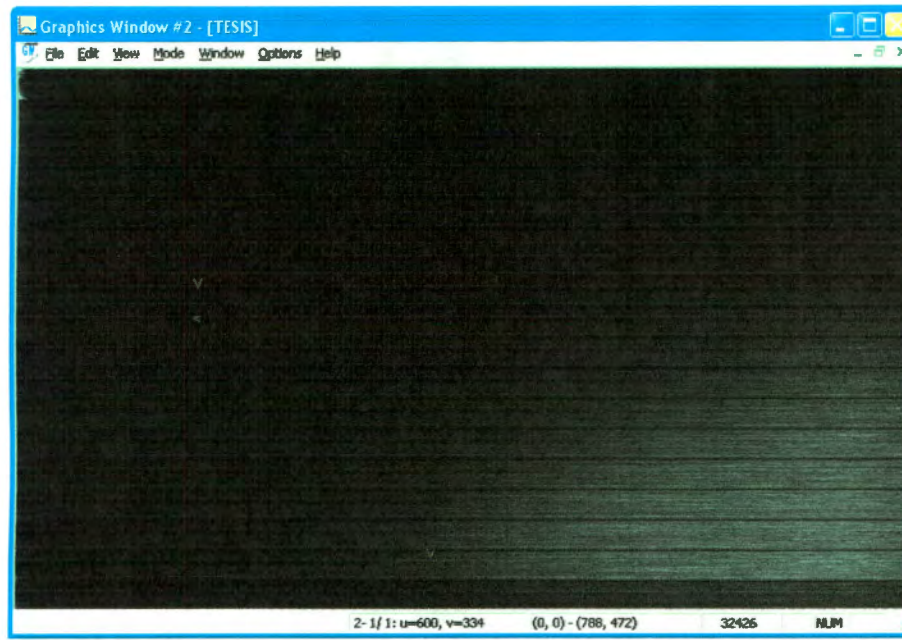


Fig. 4.20 Simulación 1000 generaciones 1000 individuos con depredador aleatorio y reducción de malla toroidal.

Como podemos observar al reducir la malla toroidal el depredador mató a 2 presas como se muestra en la fig. 4.20 y en la figura 4.19 vemos que el tiempo de vida mínimo, depredador menos inteligente, disminuyó a 1000, el tiempo de vida máximo está oscilando con tendencia hacia arriba. Ahora continuamos las pruebas cambiando el depredador aleatorio por uno constante manteniendo reducida la malla toroidal y obtuvimos lo siguiente:

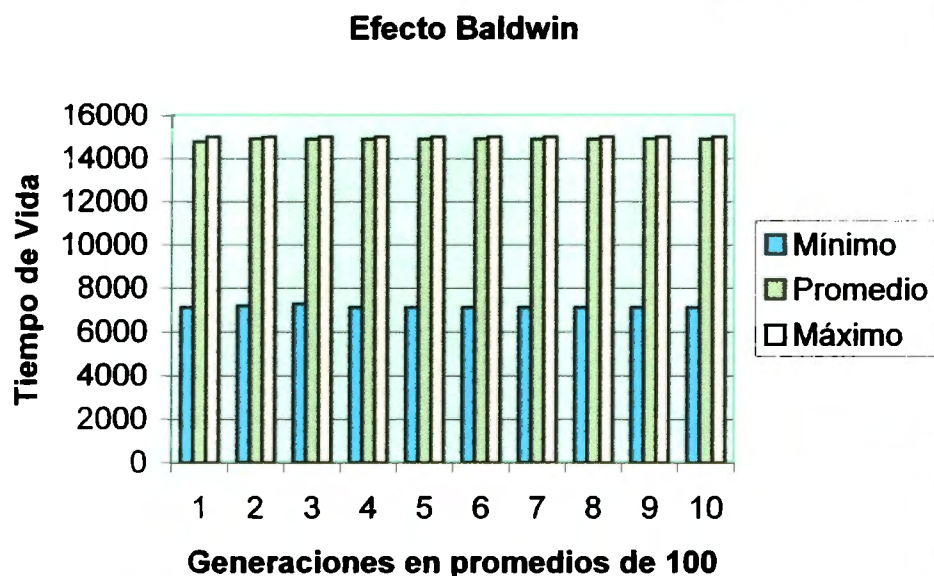


Fig. 4.21 Evolución 1000 generaciones 1000 individuos con depredador constante y reducción de malla toroidal.



Fig. 4.22 Simulación 1000 generaciones 1000 individuos con depredador constante y reducción de malla toroidal.

Como podemos observar en la figura 4.21 la tendencia es hacia arriba del tiempo de vida promedio, el tiempo de vida máximo es constante. En la figura 4.22 el depredador se come una pieza, una presa sigue la estrategia de copiar la actitud del depredador y se separa de las demás. Después continuamos haciendo pruebas, y cambiamos el depredador constante por un semi aleatorio con malla reducida y obtuvimos lo siguiente:

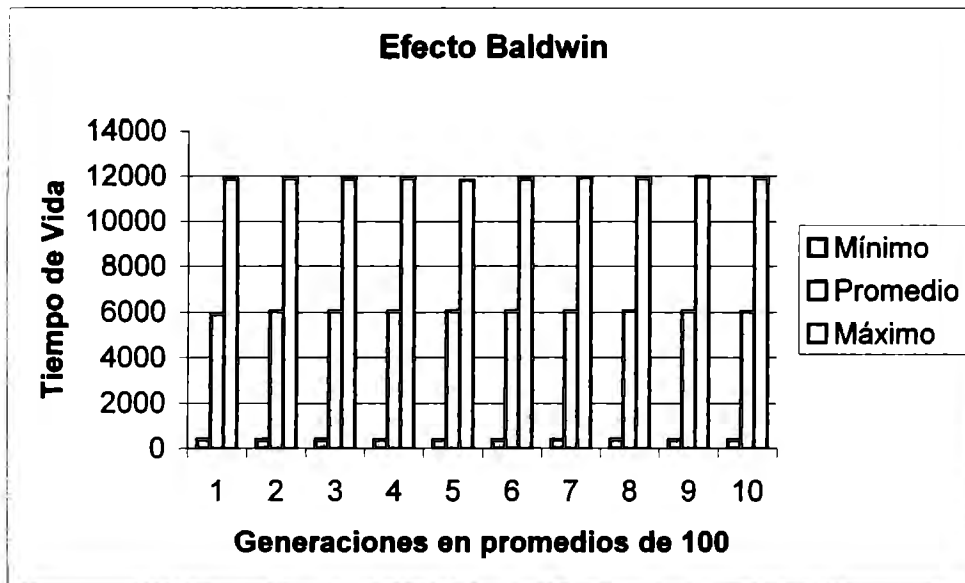


Fig. 4.23 Evolución 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de malla toroidal.

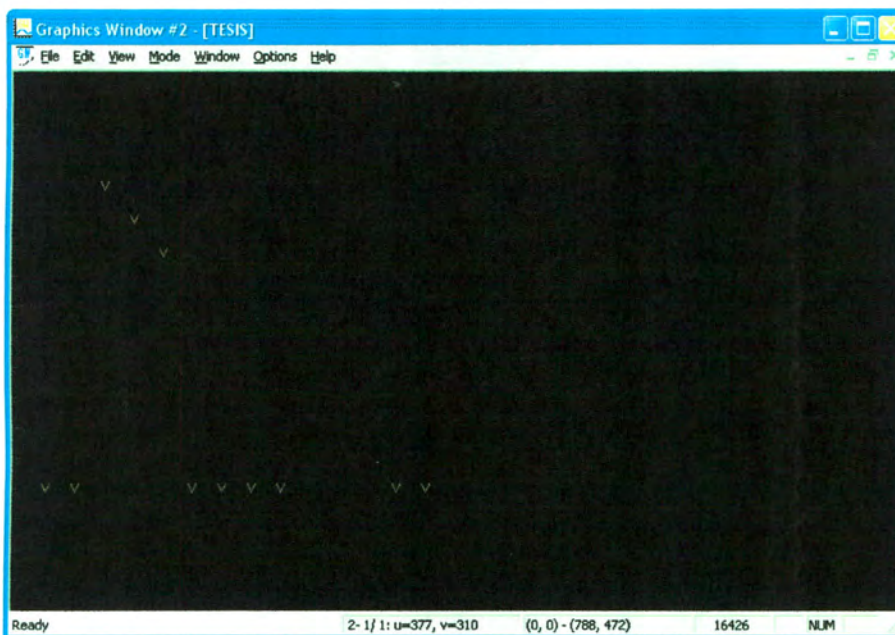


Fig. 4.24 Simulación 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción de malla toroidal.

Como podemos observar en la grafica 4.23 en la siguiente gráfica el tiempo de vida del máximo oscila hacia arriba. El promedio oscila hacia arriba, se forman grupos con mayor claridad. El depredador mata a cuatro presas.

4.3 Comparación.

La siguiente grafica es una comparación entre la figura 4.1 y la figura 4.3

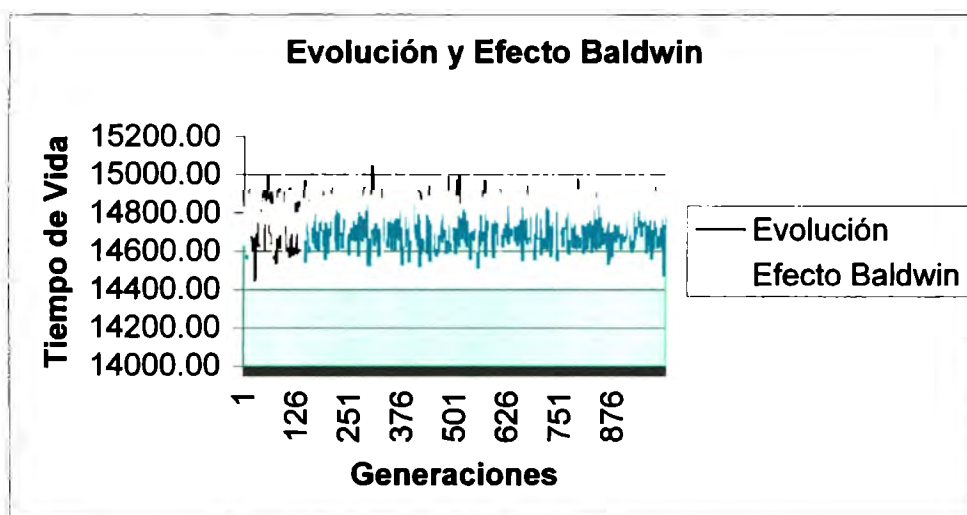


Fig. 4.25 Evolución y efecto Baldwin 1000 generaciones 100 individuos con depredador aleatorio

Como se puede observar en la figura 4.25 el efecto Baldwin oscila por arriba de la Evolución. La siguiente gráfica muestra una comparación entre la evolución y el efecto Baldwin en con un depredador aleatorio y una reducción de la malla toroidal.

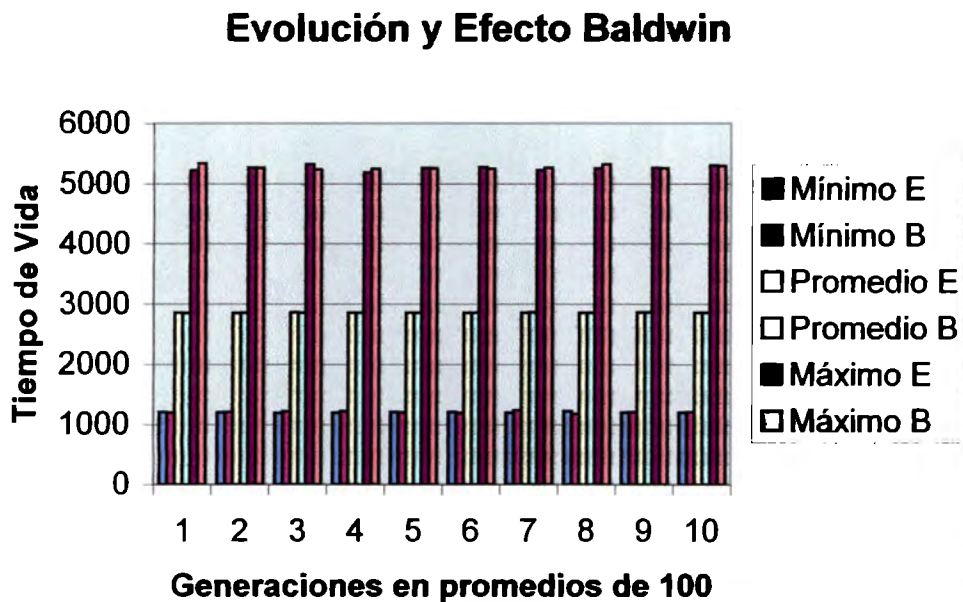


Fig. 4.26 Evolución y Efecto Baldwin 1000 generaciones con 1000 individuos con depredador aleatorio y reducción de malla toroidal.

En la figura 4.26 puede observar que cuando el tiempo de vida máximo del efecto Baldwin sube la evolución sube y cuando el tiempo de vida máximo del efecto Baldwin Baja el tiempo de vida de la evolución baja. A continuación presentamos la misma comparación pero para un depredador constante y reducción de la malla toroidal.

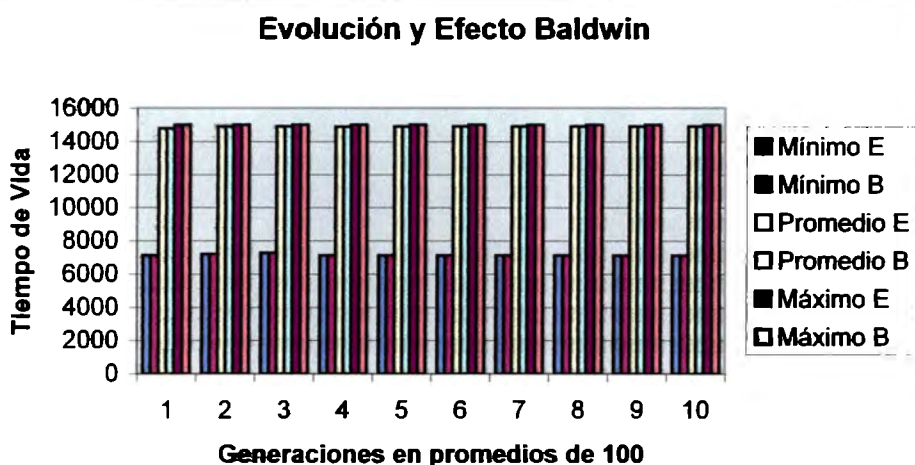


Fig. 4.27 Evolución y efecto Baldwin 1000 generaciones con 1000 individuos con depredador constante y reducción de malla toroidal.

Se puede observar en la figura 4.27 que el tiempo de vida mínimo, de las presas más tonta, incrementa y tiende hacia arriba en las primeras generaciones, luego permanece constante. A continuación se presenta la comparación del efecto Baldwin y evolución para un ambiente semi aleatorio y con reducción de la malla toroidal.

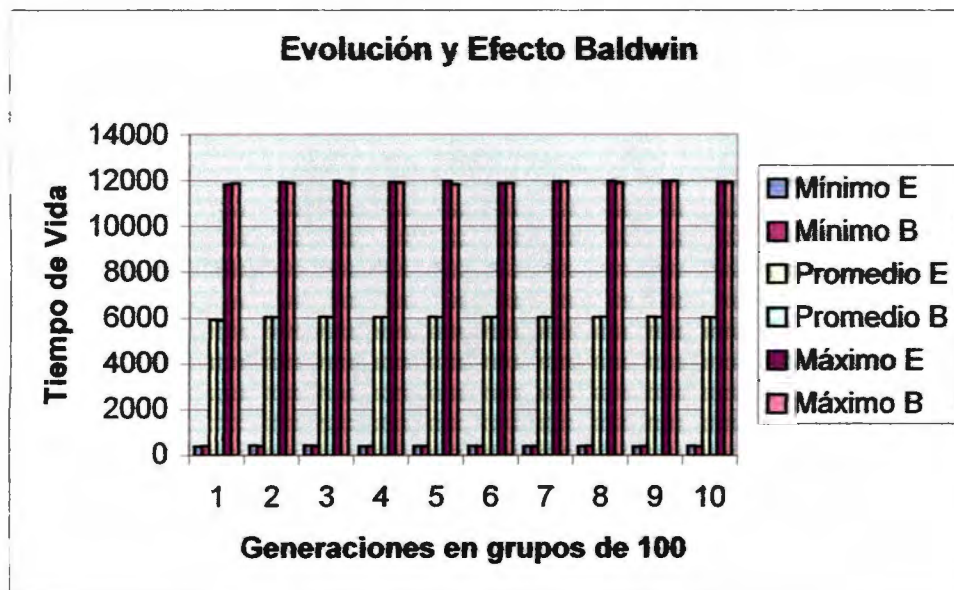


Fig. 4.28 Evolución y efecto Baldwin 1000 generaciones con 1000 individuos con depredador semi aleatorio y reducción de malla toroidal.

Como podemos observar en la figura el tiempo de vida máximo y el tiempo de vida mínimo, oscila hacia arriba. Finalmente se presenta una comparación en la modificación de los pesos entre el efecto Baldwin y la Evolución en el depredador semialeatorio y reducción de la malla toroidal.

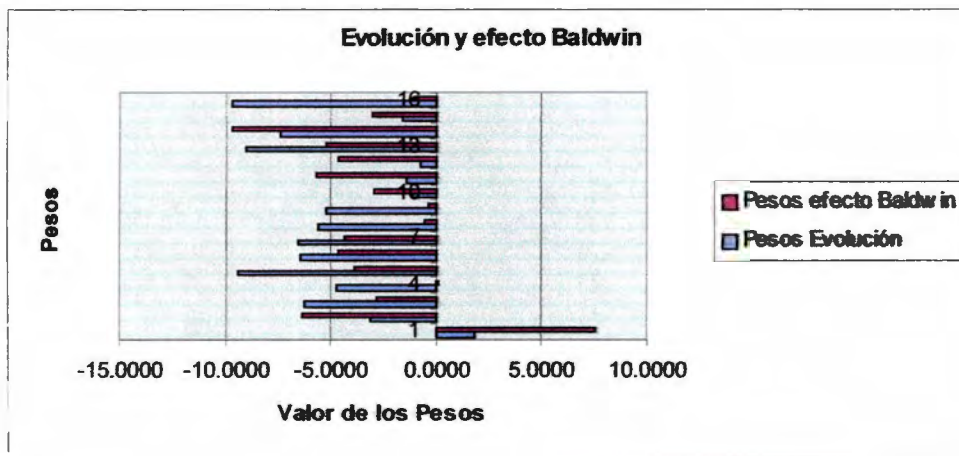


Fig. 4.29 Pesos evolución y efecto Baldwin 1000 generaciones 1000 individuos con depredador semi aleatorio y reducción malla toroidal

Como podemos observar en la gráfica 4.29 la mayoría de los pesos con el efecto Baldwin disminuyen con respecto a la pura evolución.

A continuación se muestra una gráfica que muestra de la generación 665 a la 672 con malla toroidal y tomando en cuenta la presa más apta o el tiempo de vida máximo.

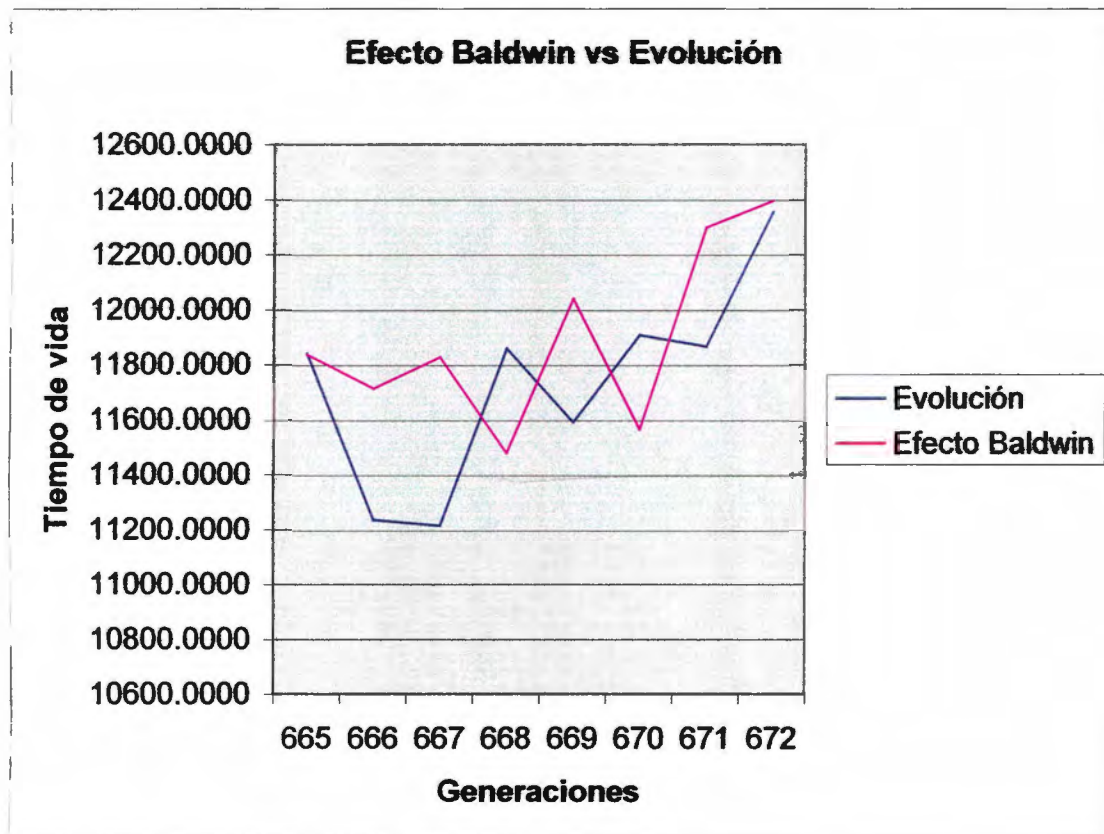


Fig. 4.30 Evolución y efecto Baldwin 1000 generaciones 1000 individuos con depredador semi aleatorio con elitismo y reducción de la malla toroidal.

En esta gráfica observamos que ambas líneas oscilan hacia arriba. En la mitad de la generación 666 el efecto Baldwin empieza a subir e inmediatamente a la mitad de la generación 667 la evolución empieza a subir. Ahora la evolución empieza a bajar y en la generación 668 la evolución empieza a bajar. Después en lo que ahora el efecto Baldwin empieza a subir y en la generación 669 la evolución empieza a subir. Observamos que ambas líneas siguen un comportamiento hacia arriba y la evolución cambia de comportamiento según el comportamiento del efecto Baldwin.

5. CONCLUSIONES

Los resultados muestran que en un ambiente muy dinámico o aleatorio es muy difícil para que el aprendizaje y la evolución puedan manifestarse cabalmente como fenómenos de adaptación eficaces, lo que ha sido reportado en trabajos previos de Littman. [11][13] Las presas no muestran un comportamiento apropiado para el medio ambiente, lo que provoca que mueran rápidamente.

En estos trabajos, el aprendizaje resulta ser un proceso muy complejo para las presas ya que por lo general son atrapadas por el depredador a medida que transcurre el aprendizaje. Considero que esto nos dice que un ambiente muy dinámico provoca aleatoriedad en los comportamientos dada la incertidumbre de las presas y poco poder de adaptación, pues no logran aprender ni evolucionar hacia una misma dirección por lo que este caso es el peor para las presas pues todas mueren. (ver fig. 4.8.) Dados esos cambios tan drásticos tenemos como resultado que un mismo aprendizaje puede a veces estar correcto en un ambiente e incorrecto en otro.

En cambio, si las presas se ubican en un ambiente constante observamos que el nivel de adaptación de las presas se eleva a niveles óptimos hasta que el aprendizaje deja de ser un elemento esencial para su supervivencia. Como consecuencia se deja de aprender. (ver figura 4.27.) dado que el depredador constante llega un momento en que no persigue a las presas como se puede observar en la figura 4.22.

Como también se puede observar en la fig. 4.16, el depredador constante es el que va en la misma dirección. Esto provoca que las presas aprendan muy poco. El depredador sólo mata a la única presa que está en su dirección, esto es relevante para demostrar que un ambiente constante hace que el aprendizaje de las presas tienda a disminuir y por efecto Baldwin de evolucionar. Esto nos dice que un ambiente en donde todo es constante a una tendencia más alta a la perfección y que el aprendizaje guía la evolución.

Los resultados muestran que un ambiente semi aleatorio o semi dinámico las presas logran adaptarse más que en un ambiente aleatorio pues siempre continúan aprendiendo. Se puede ver en la figura 4.30 el efecto Baldwin que está guiando la evolución mejor que en un sistema aleatorio. Los resultados muestran que las presas empiezan a crear un comportamiento emergente que es agruparse[24] este comportamiento emergente es resultado y no prefabricado a diferencia del trabajo de Craig Reynolds[4] y las presas se agrupan más en efecto Baldwin que en evolución. Ver fig 4.12 y 4.24.

Utilizar un depredador semi aleatorio es la mejor opción para que las presas aprendan y evolucionen. Reducir la malla toroidal favorece a la entropía, es decir, provoca que las presas se motiven para sobrevivir y a aprender. En otras palabras disminuir dicha malla hace que el depredador mate más presas lo cual obliga a las presas a aprender y evolucionar. Observamos que la presa se puede adaptar más que en un sistema aleatorio. Bajo este esquema se ayuda a las presas a que sean perfectas en más tipos de ambientes y no solamente en uno. Observamos que cuando existe mayor presión las presas tienden a unirse. El tiempo de vida de la presa siempre tiende a subir.

El tiempo de vida disminuye cuando hay un cambio muy drástico en el ambiente, y cuando por aleatoriedad el ambiente mata a las presas sin explicación alguna como es en el caso aleatorio. Lo que se ve en las gráficas es que aunque cae el tiempo de vida debido al ambiente, la evolución y efecto Baldwin(evolución con aprendizaje) tienden hacia arriba. Pero esta última relativamente más que la evolución sin aprendizaje. Es interesante observar el efecto Baldwin y la evolución aquí vemos que existe otro punto a favor de que el aprendizaje guía la evolución.

Es interesante observar que en la figura 4.30 podemos interpretar que evolucionar y aprender es mejor que evolucionar en la mayoría de los casos. Pero hay circunstancias en donde lo aprendido puede resultar útil para un ambiente anterior y resultar inútil o malo para el ambiente siguiente en que uno está lo cual provoca que halla una caída en el tiempo de vida pues el aprendizaje no es el correcto para el nuevo ambiente.

La evolución es un proceso natural que está basado en un proceso de selección, cruzamiento y mutación. Los resultados muestran que el tiempo de vida del sistema va incrementando a medida que van evolucionando los individuos del algoritmo genético. También se observa que la selección es una parte importante del proceso evolutivo porque es la que decide la optimización. Es curioso observar que en dicha optimización influye también el medio ambiente dinámico en el que se sitúa un cromosoma. [5]

La evolución con aprendizaje es un proceso natural más tardado que también se basa en un proceso de selección con aprendizaje, cruzamiento y mutación. La diferencia de agregar aprendizaje muestra en los resultados un incremento mayor en el tiempo de vida del sistema a medida que van evolucionando los individuos del algoritmo genético. También se observa que la selección es una parte importante del proceso evolutivo con aprendizaje porque es la que decide la optimización. [5][7]

El uso de algoritmos genéticos y aprendizaje para optimizar los pesos de una red neuronal sirven para un simulador depredador presa basado en el efecto Baldwin porque con la utilización de dichas técnicas computacionales pudimos comprobar la hipótesis de que el aprendizaje guía la evolución.

Es muy importante observar que en el Baldwin el aprendizaje afecta en el proceso de selección más no en los cromosomas de las presas. Parafraseando, el aprendizaje afecta en el tiempo de vida del sistema, y en los pesos de la red neuronal de las presas, generados por los cromosomas pero no en los cromosomas mismos. Dado que el proceso de selección está basado en el tiempo de vida de las presas y por lo tanto tiempo de vida del sistema concluimos que el aprendizaje modifica los tiempos de vida sin alterar el material genético por lo que indirectamente está afectando la evolución.

El algoritmo de evolución con aprendizaje es un proceso más lento que el algoritmo de evolución porque el algoritmo de evolución duró seis horas y el algoritmo de evolución con aprendizaje duró siete horas. Los resultados muestran que las presas tienden a estar en grupos para evadir a su depredador.

Una limitación de nuestro ambiente que es el depredador no evoluciona pues utiliza un algoritmo para cazar aleatorio, semi-aleatorio o constante programado y no mediante un algoritmo evolutivo. Un sistema en donde exista coevolución podría ser más enriquecedor.

Otra limitación de nuestro sistema es que tanto las presas como el depredador solamente se pueden mover en direcciones norte, sur, este, oeste por lo que también sería mejor en trabajos futuros permitirle a las presas y al depredador un espacio de movimiento en diagonal y ver que ocurre cuando existen más incrementan las posibilidades de acción de las presas y el depredador.

Otra limitación de nuestro sistema es que solamente utilizamos una red neuronal con cinco entradas y dos neuronas en la capa intermedia, y tres neuronas en la capa de salida para cubrir las percepciones, y los movimientos de las presas. También sería enriquecedor aumentar el número de percepciones de la presa o depredador y observar que pasa cuando se tienen más posibilidades para actuar y más posibilidades para percibir.

A diferencia de los modelos de Hinton y Nowlan, en este modelo el medio ambiente es importante en el proceso de selección y el aprendizaje resulta de un esquema basado en la vida real y comprobado por Craig Reynolds[4], y no de un algoritmo aleatorio. [10]

Diferente a los modelos de Ackley y Littman, nuestro modelo trata la evolución y la evolución con aprendizaje para después observar lo que pasa en un simulador depredador presa. En los resultados se muestra también que el aprendizaje influye indirectamente en la evolución. Parafraseando, el aprendizaje permite que la evolución siga su curso pues interviene en ella indirectamente en el proceso de selección. [3]

En contraste con el trabajo de Nishima y Takashi pudimos agregar aprendizaje al modelo, y observar que las presas tienden efectivamente a agruparse en situaciones de incertidumbre para sobrevivir. Comprobamos que un sistema toroidal es factible para dar grados de libertad a un sistema en donde el ambiente dinámico, permite a las presas desenvolverse ampliamente y generar el comportamiento emergente que se observó. Ver figura simulaciones semi-aleatorio en donde mientras más aprenden tiende emergentemente a juntarse. [24]

Como se puede observar los tiempos de vida generados por el efecto Baldwin son más altos lo que nos comprueba la hipótesis de que la evolución con aprendizaje es mejor que la evolución pura pues el aprendizaje guía la evolución. Por los resultados obtenidos, la evolución con aprendizaje es más tardada porque duró una hora más que la evolución pero la primera es más efectiva porque el tiempo de vida de las presas y del sistema es mayor, generando así presas con mayor aptitud y desenvolvimiento en el sistema.

Como pudimos observar el delta o el diferencial en el tiempo de vida de la evolución con respecto a la evolución con aprendizaje la constante de aprendizaje se estableció en niveles muy bajos intencionalmente con el propósito de comprobar si niveles bajos de aprendizaje tienen una influencia considerable en el tiempo de vida y así comprobar nuestra hipótesis de que el aprendizaje guía la evolución o efecto Baldwin.

También observamos con las gráficas que el ambiente, en este caso el depredador, influye en el proceso de selección y comprendemos que los estudios previos del efecto Baldwin tienen

limitaciones como el no contemplar el medio ambiente dinámico como un factor importante, ni analizar el comportamiento emergente y las implicaciones que resultan del mismo en el ambiente.

Se observó el tiempo de vida, las acciones y el comportamiento de las presas generados por el proceso evolutivo, el proceso evolutivo con aprendizaje y verifico la teoría del efecto Baldwin en un simulador depredador presa artificial.

Con estudios pudimos observar los patrones que siguen los seres vivos en un simulador depredador presa para comportarse son el ambiente y el aprendizaje, así como la evolución. La importancia de este proyecto consistió en comprobar que existen modelos computacionales basados en vida artificial que pueden ayudarnos a optimizar procesos de búsqueda o selección de acciones así pues como al entendimiento de fenómenos biológicos como el comportamiento, el aprendizaje y la evolución.

En conclusión, hemos comprobado la hipótesis solamente para un sistema depredador presa de que el aprendizaje guía la evolución sin cambiar el material genético con el que se nace. Comprobamos también que la selección del mejor esta dada por aquel que tiene un tiempo de vida más alto y si el aprendizaje ayuda a que se tenga un tiempo de vida mayor entonces podemos decir que el aprendizaje afecta ese mecanismo de decisión que guía la selección en los procesos evolutivos. Esta argumentación coincide con la presentada previamente por Craig Reynolds. [4]

Durante los experimentos preliminares se aplicó la metodología, propuesta por Nolfi[13] que es tomar el aprendizaje por separado una vez que la evolución había concluido. Esto hacía que el aprendizaje casi nulo y no afectaba en lo absoluto en la evolución. Se modificaron los programas para que se integrara el aprendizaje en la función de aptitud para simular el efecto Baldwin. Parafraseando, el aprendizaje se integra en la función de aptitud y no por separado de la evolución si se quiere simular el efecto que Baldwin propone.

Para trabajos futuros recomendamos aumentar las percepciones de las presas, o aumentar las percepciones del depredador, coevolucionar el depredador con las presas y analizar resultados emergentes a medida que se van agregando movimientos simples. Interesante podría ser agregar otros algoritmos de aprendizaje en la función de aptitud y observar diferencias entre tipos de aprendizaje. Analizar que tanto influye la evolución en las estrategias de aprendizaje si es que existe alguna correlación.

6 REFERENCIAS Y BIBLIOGRAFÍA.

- [1] MITCHELL, Mellanie, *An Introduction to Genetic Algorithms*. MIT Press edition metric edition London, England: MIT, 1999. p. 208.
- [2] BELEW, Richard ET MITCHELL, Mellanie. *Adaptive Individuals in Evolving Populations Models and Algorithms*. Adaptado por D. Bacon .Addison-Wesley London, England: MIT, 1996 p.531.
- [3] NOLFI, Estefano, FLOREANO, Darío, BELEW, Richard. *Evolutionary Robotics*. MIT, 2000 p. 320.
- [4] REYNOLDS, Craig, An Evolved, Vision-Based Behavioral Model of Coordinated Group Motion, ARCADY JEAN, Meyer, WILSON, Stewart.(Eds) En *From Animats to Animats 2 Proceedings of the second international conference on Simulation of Adaptive Behavior*. San Mateo California,USA p.384-292.
- [5] ALCOCK, John. *Animal Behavior* . Sixth edition, Sinauer Associates, Massachussets, 1998. p. 640
- [6] LAMARCK J.B. Of the Influence of the Environment on the Activities and Habits of these hung bodies on Modifying their organization and structure” En BELEW, Richard, MITCHELL Mellanie(Eds), *Adaptive Individuals in Evolving Populations Models and Algorithms*. Adaptado por D. Bacon .Addison-Wesley London, England: MIT, 1996. p.33
- [7] BALDWIN, J. M. A new Factor in Evolution En BELEW, Richard, MITCHELL Mellanie (Eds)., *Adaptive Individuals in Evolving Populations Models and Algorithms*. Adaptado por D. Bacon .Addison-Wesley London, England: MIT, 1996. p.59
- [8] SIMPSON, G.G. The Baldwin Effect En BELEW, Richard, MITCHELL Mellanie (Eds)., *Adaptive Individuals in Evolving Populations Models and Algorithms*. Adaptado por D. Bacon .Addison-Wesley London, England: MIT, 1996. p.99.
- [9] PARISI and NOLFI The influence of Learning on Evolution En BELEW,Richard, MITCHELL Mellanie (Eds)., *Adaptive Individuals in Evolving Populations Models and Algorithms*. Adaptado por D. Bacon .Addison-Wesley London, England: MIT, 1996 p 419
- [10] HINTON G.E. and NOWLAN S.J. How learning can guide Evolution En BELEW,Richard, MITCHELL (Eds) Mellanie., *Adaptive Individuals in Evolving Populations Models and Algorithms*. Adaptado por D. Bacon .Addison-Wesley London, England: MIT, 1996. p.447.
- [11] M.L LITTMAN Simulations Combining Evolution and Learning En BELEW,Richard, MITCHELL Mellanie. (Eds), *Adaptive Individuals in Evolving Populations Models and Algorithms*. Adaptado por D. Bacon .Addison-Wesley London, England: MIT, 1996 p.461.
- [12] LANGTON, Christofer., *Artificial Life II a Proceeding Volume X in Santa Fé Institute studies in the Sciences of Complexity*. Addison-Wesley, California, 1992. p.855.
- [13] ACKLEY, David et LITTMAN,Michael Interactions between learning and Evolution En LANGTON,Christofer.(Ed), *Artificial Life II a Proceeding Volume X in Santa Fé Institute studies in the Sciences of Complexity*. Addison-Wesley, California, 1992. p.487
- [14] BELEW, Richard et MCINERGY, John Evolving Networks using the genetic Algorithm with conexionist Learning En LANGTON, Christofer.(Ed), *Artificial Life II a Proceeding*

- Volume X in Santa Fé Institute studies in the Sciences of Complexity*. Addison-Wesley, California, 1992. p.511.
- [15] KOZA, John Genetic Evolution and Co-Evolution of Computer Programs En LANGTON, Christofer.,(Ed) *Artificial Life II a Proceeding Volume X in Santa Fé Institute studies in the Sciences of Complexity*. Addison-Wesley, California, 1992. p 549.
- [16] DORIGO, Marco ET COLOBETTI, Marco, *An Experiment in Behavior Engineering*, MIT Press, London, England, 1998.
- [17] FREEMAN, James A. ET SKAPURA, David *Neural Networks Algorithms, Applications and Programming Techniques Computation and neural systems series*, Addison-Wesley, 1992 p.401.
- [18] PFEIFER, Rolt ET SCHEIER, Christian, *Understanding Intelligence*, MIT Press, Cambridge Massachussets, 1999, p 697.
- [19] GALVAN, Blas et GARCIA, *Bartolomé Técnicas de análisis de fiabilidad y métodos evolutivos de optimización global aplicados al diseño de software*. España, p. 17
- [20] TETTAMAZI, A ET TOMASSINI, M. *Soft Computing Integrating Evolutionary Neural Fuzzy Systems* Springer, Italia, pp.62-70
- [21] SIPPER, Moshe, *An Introduction to Artificial life*, Swiss Federal Institute Im Ecublens , Switzerland pp. 1-7
- [22] DURO, Richard, *Biologically Inspired Robot Behavior Engineering* Physica-Verlag Company, España, p.438.
- [23] <http://www.dontveter.com/bpr/public2.html>
- [24] TAKASHI, Ikegami ET Shin I. Nishimura, *Emergence of Collective Strategies in a Prey-Predator Game Model*, MIT press, 1997 pp. 243 – 260.
- [25] HERTZ, J ET A. Krogh, R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison Wesley, Redwood city, CA, 1993.

ANEXOS

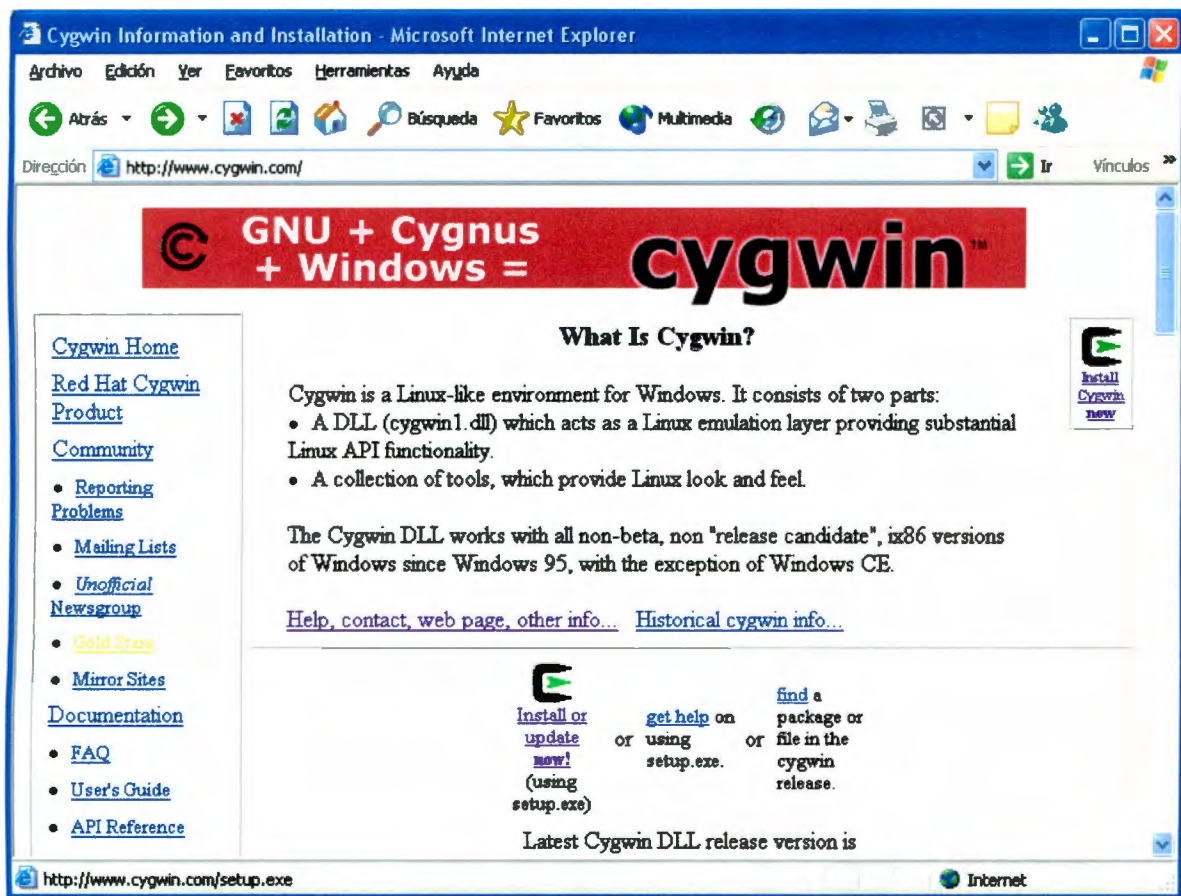
MANUAL DE INSTALACIÓN

A) REQUERIMIENTOS TÉCNICOS

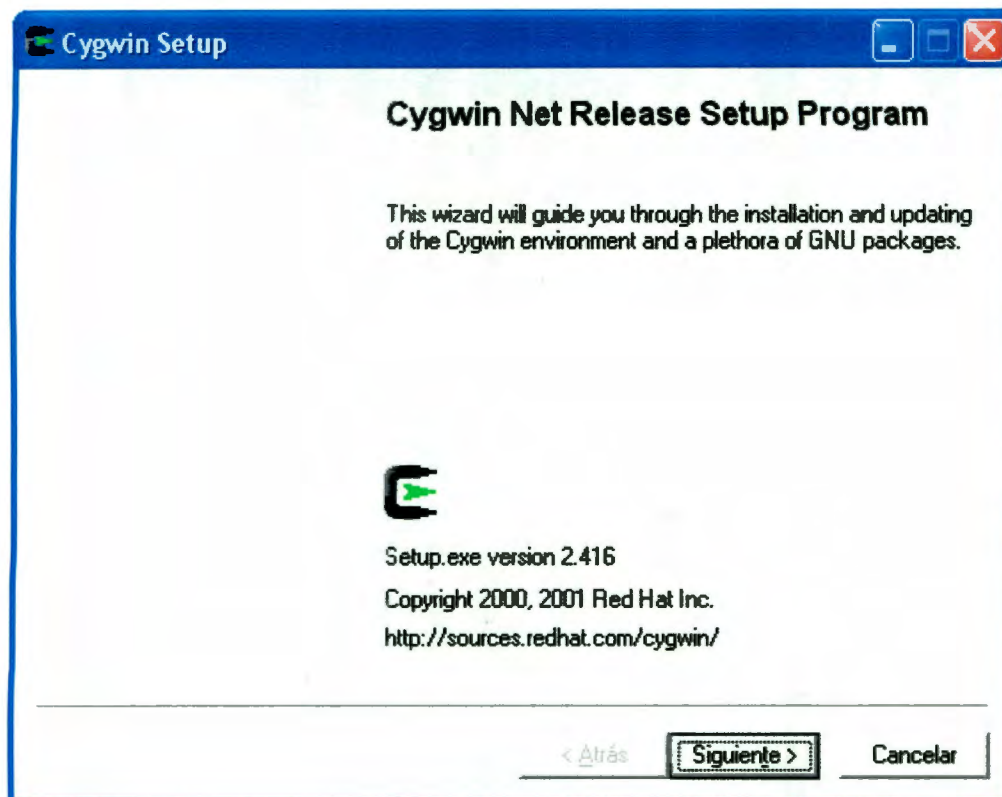
Se necesita una conexión a Internet

B) INSTALAR SIMULADOR DE LINUX PARA WINDOWS

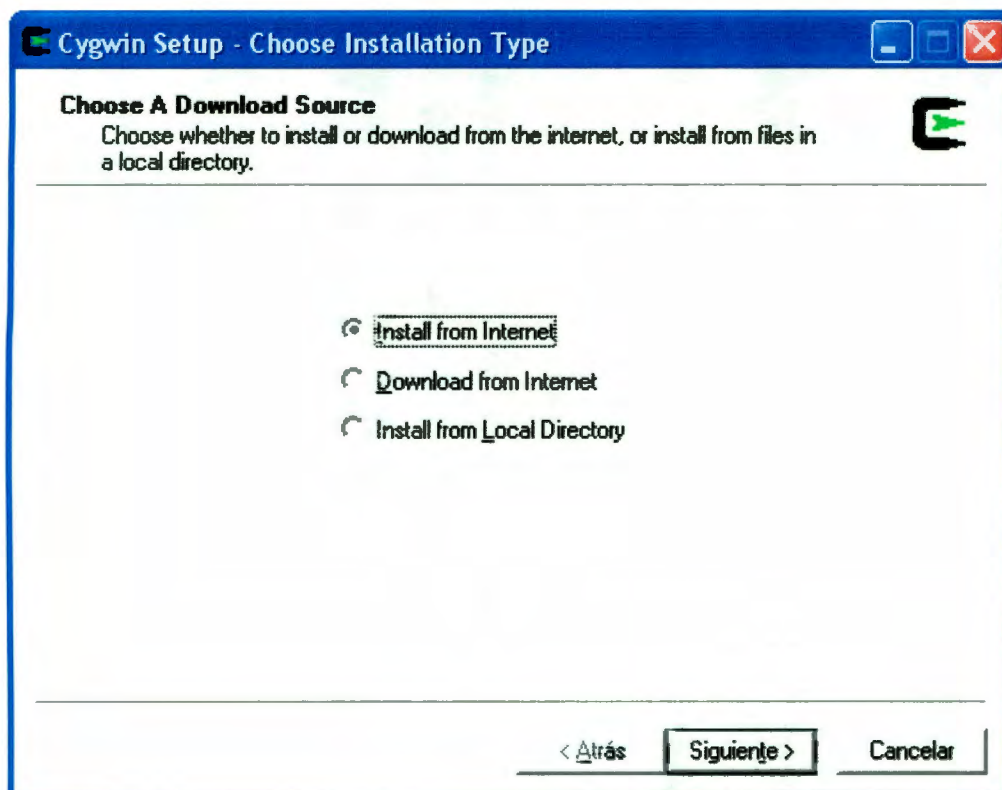
<http://www.cygwin.com>



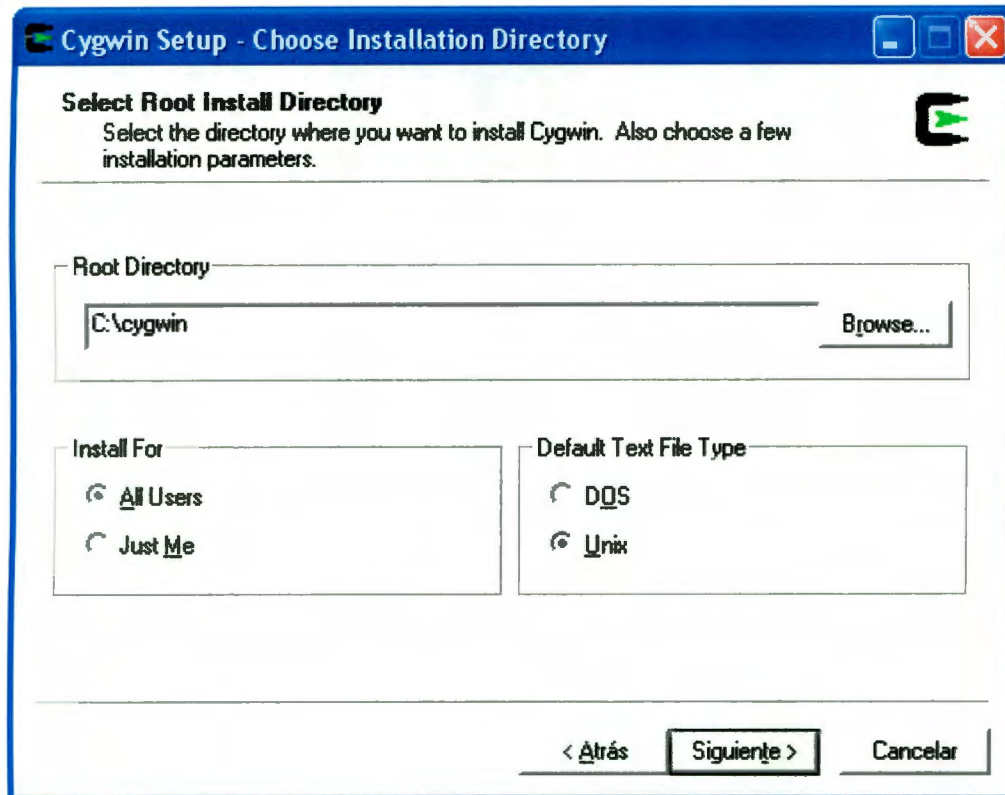
1. Copiar o bajar de esta página al escritorio el archivo del Setup.exe que también está en el DISCO
2. Dar doble clic en el archivo Setup.exe



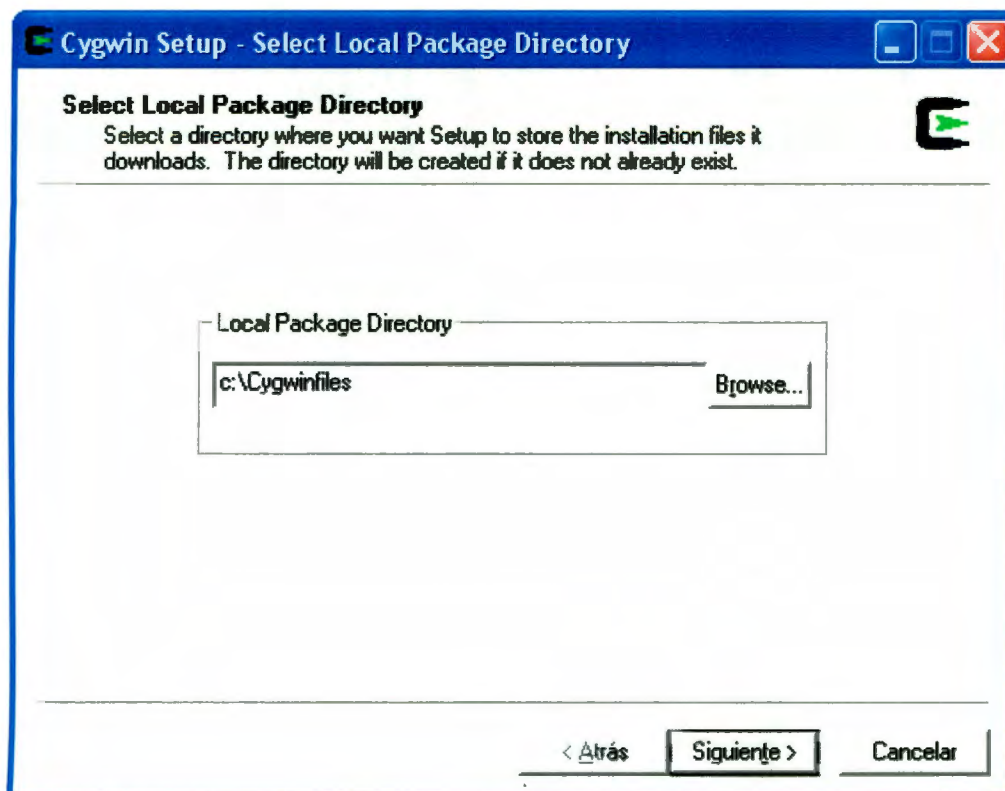
3. Hacer clic en "Siguiente"



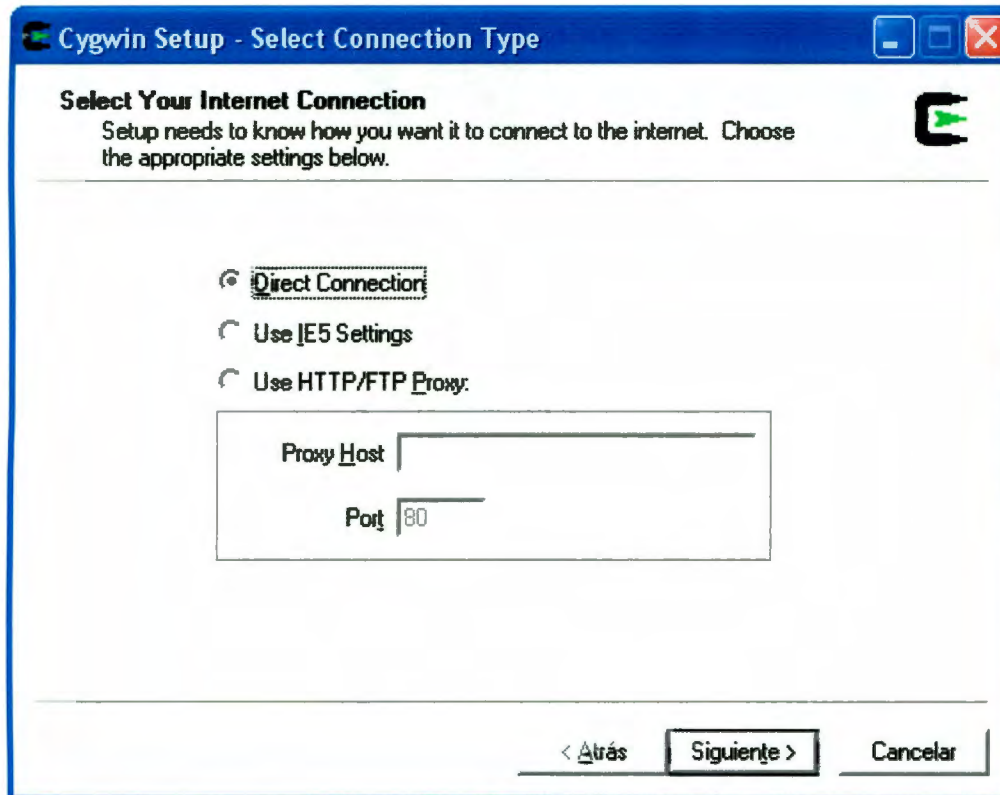
4. Seleccionar "Install from Internet" v hacer clic en "Siguiente"



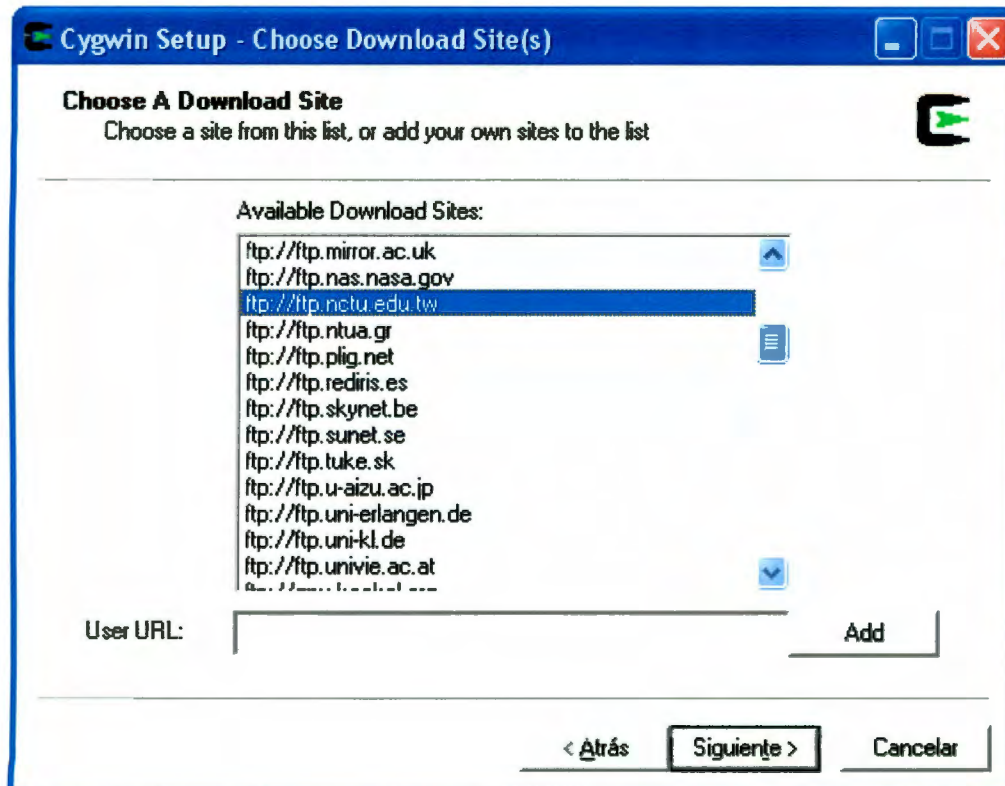
5. Seleccionar el Fólder donde se instalaran los archivos en este caso c:\Cygwin. seleccionar las propiedades "All Users" y "Unix" después presionar "Siguiete"



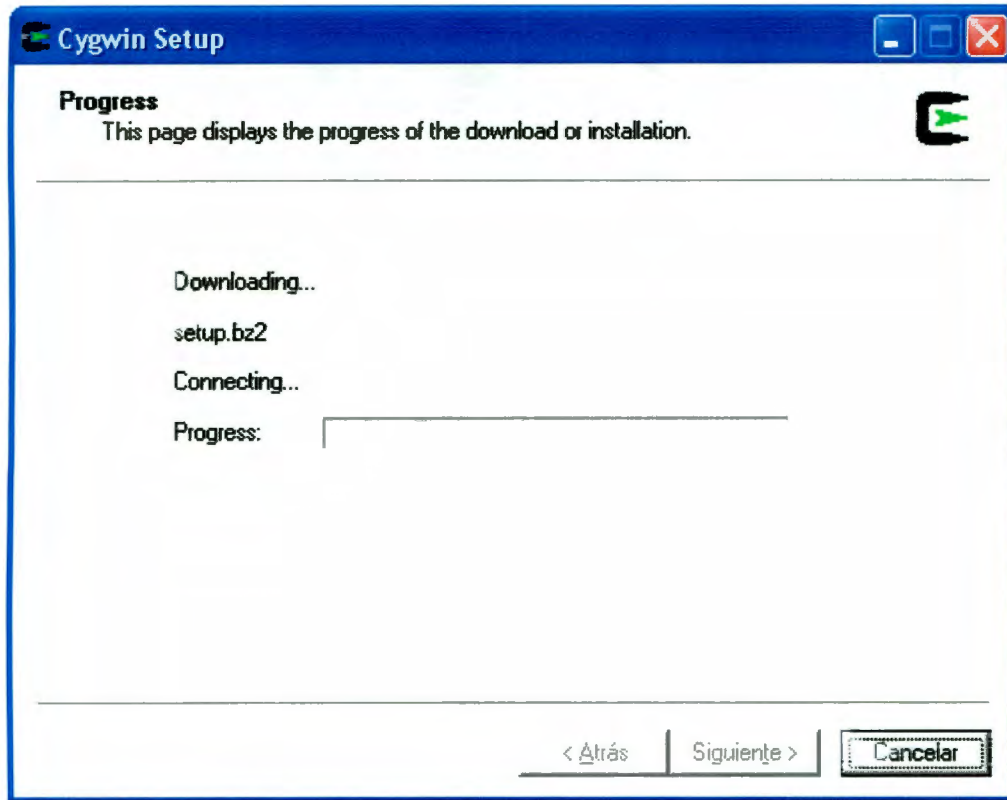
6. Seleccionar el Directorio donde se guardara la imagen de los archivos y teclear “Siguiente”



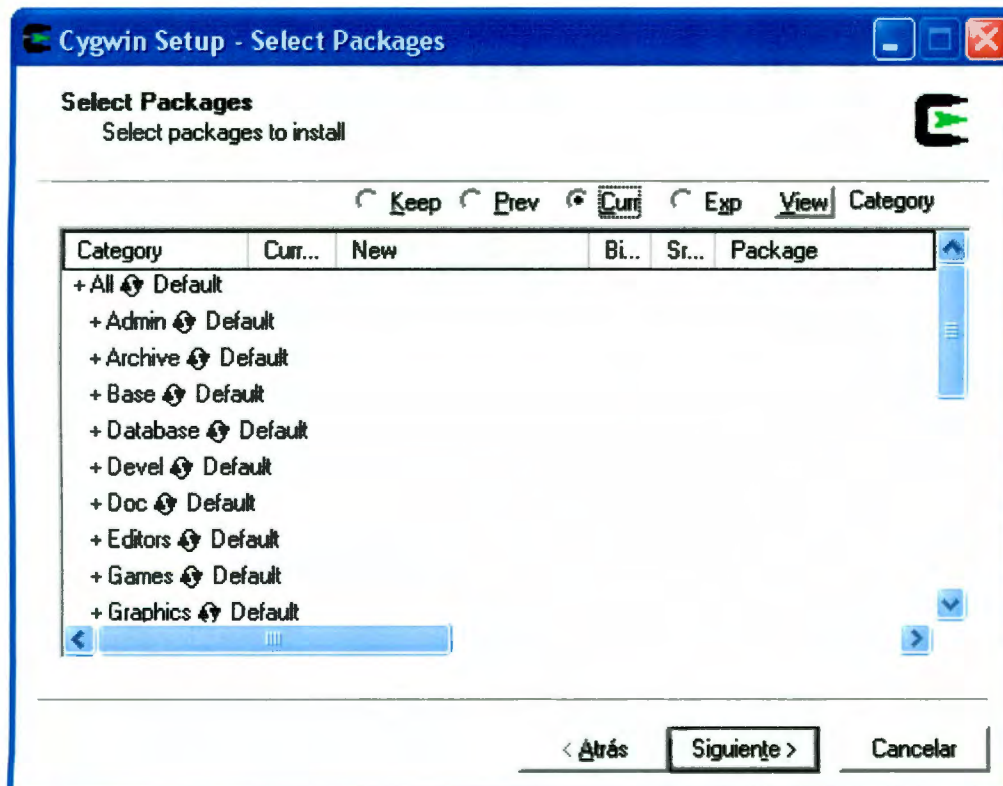
7. Seleccionar la opción que aplique y teclear siguiente:



8. Seleccionar la imagen y teclear "Siguiente"



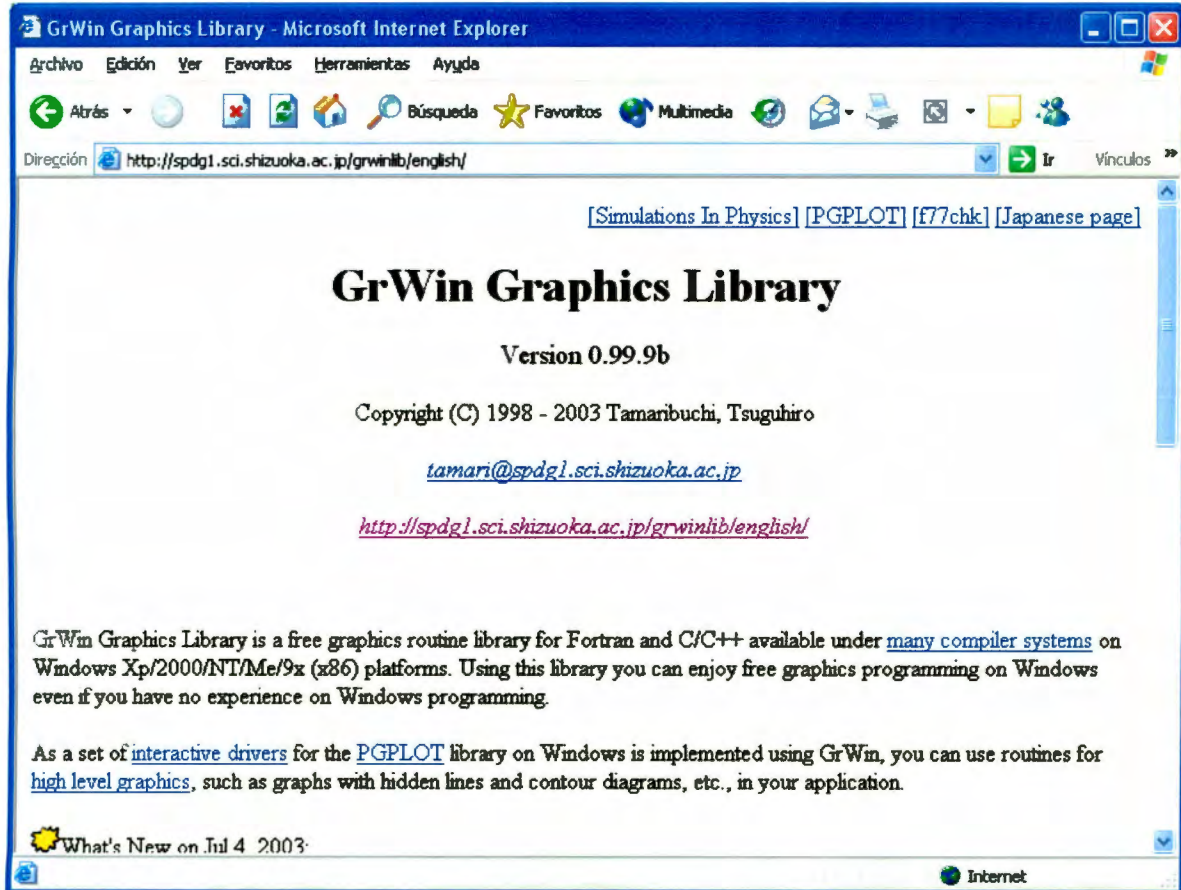
9. Esperar a que aparezca el software



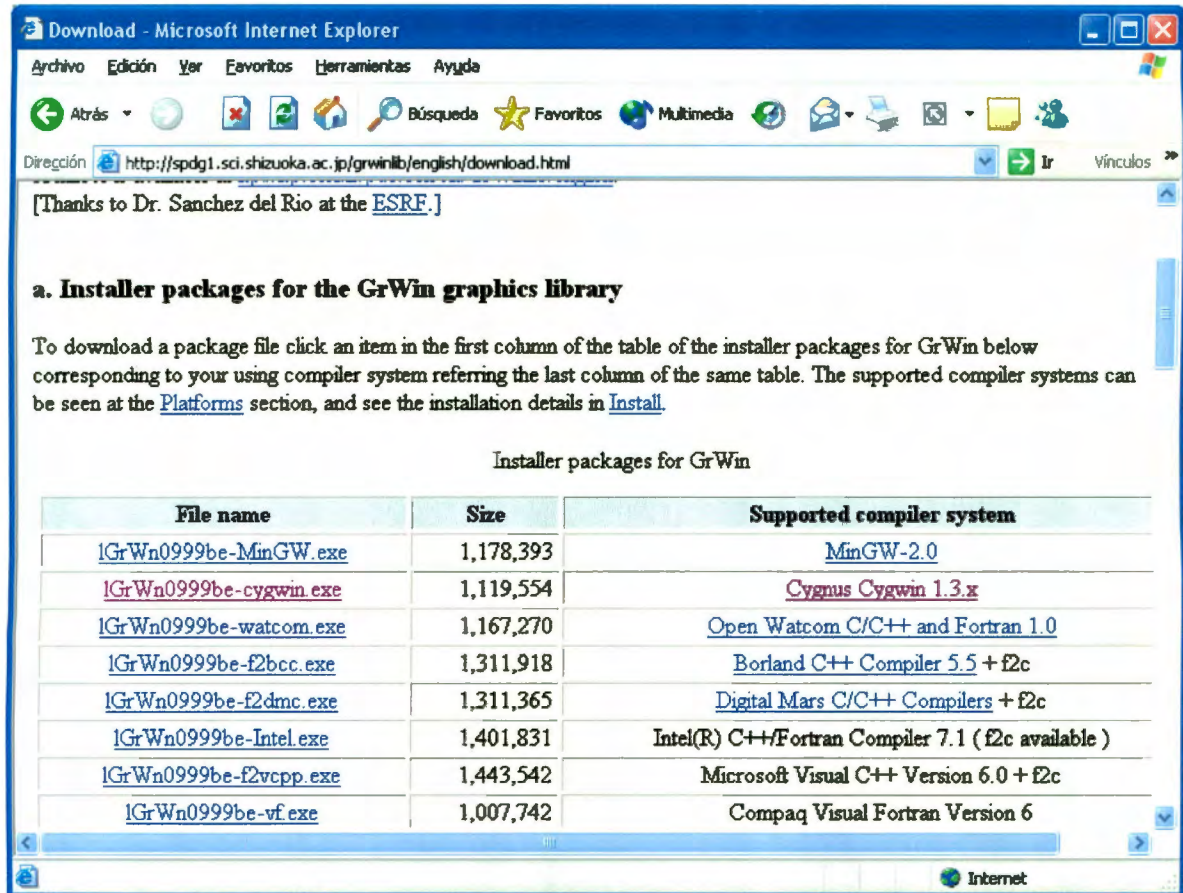
10. Instalar todo el software

C) INSTALAR LAS LIBRERIAS DEL AMBIENTE GRAFICO

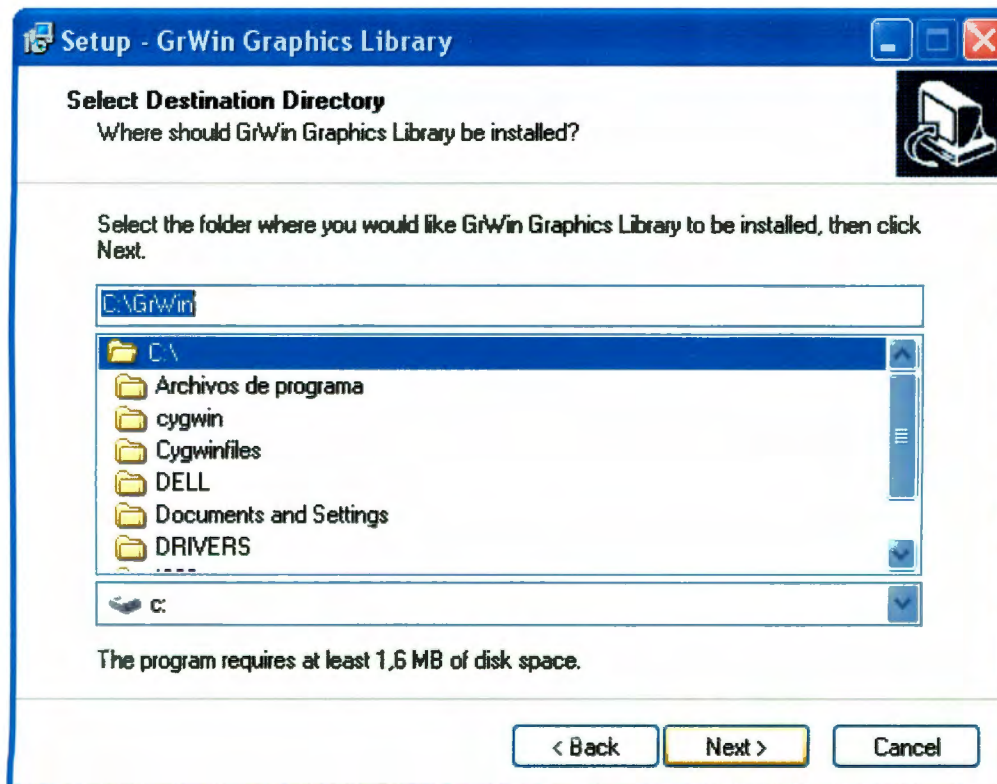
<http://spdg1.sci.shizuoka.ac.jp/grwinlib/english/>



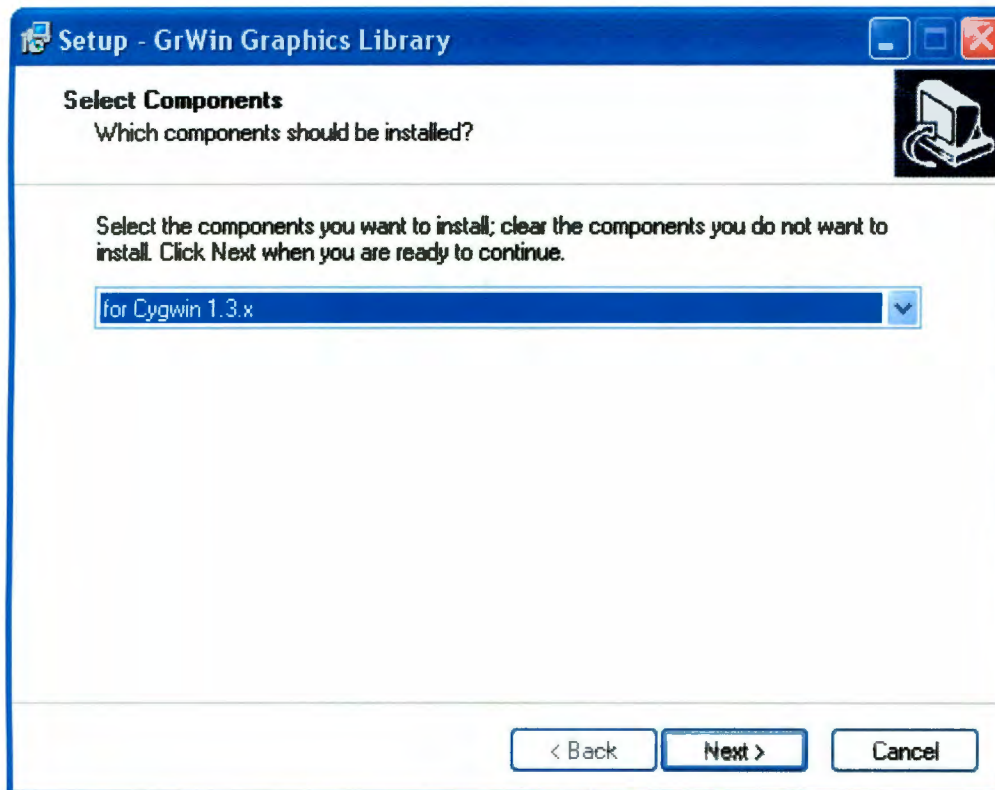
11. Seleccionar el que aplique en este caso el IGrwin0099be-cygwin.exe : en el directorio Instalación existe una copia de este ejecutable.



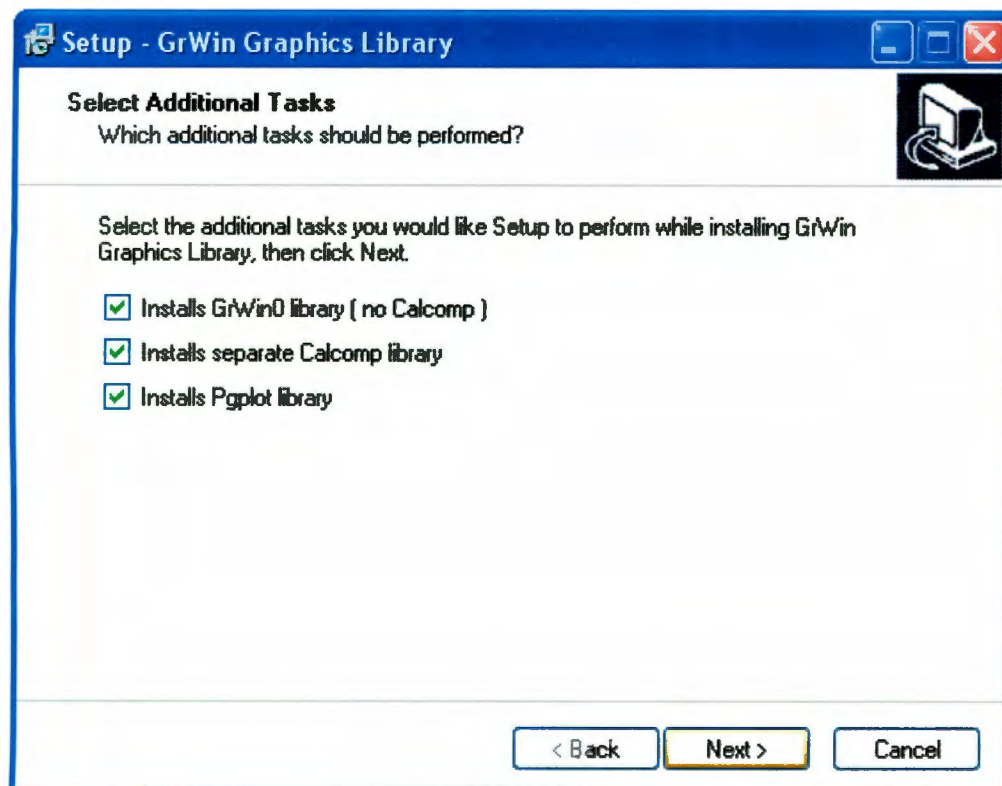
12. Dar doble clic en el instalador de la librería de Gráficos y seguir el instalador



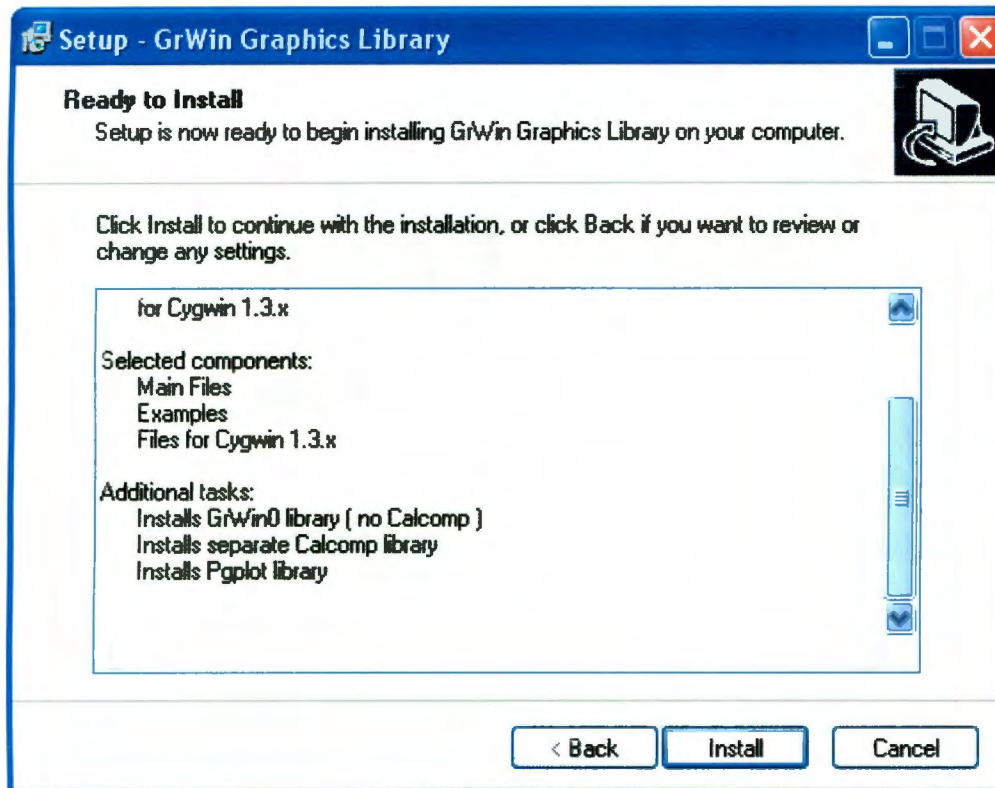
13. Seleccionar el f6lder donde se instalar6 la librer6a de gr6ficos en este caso C:\GrWin



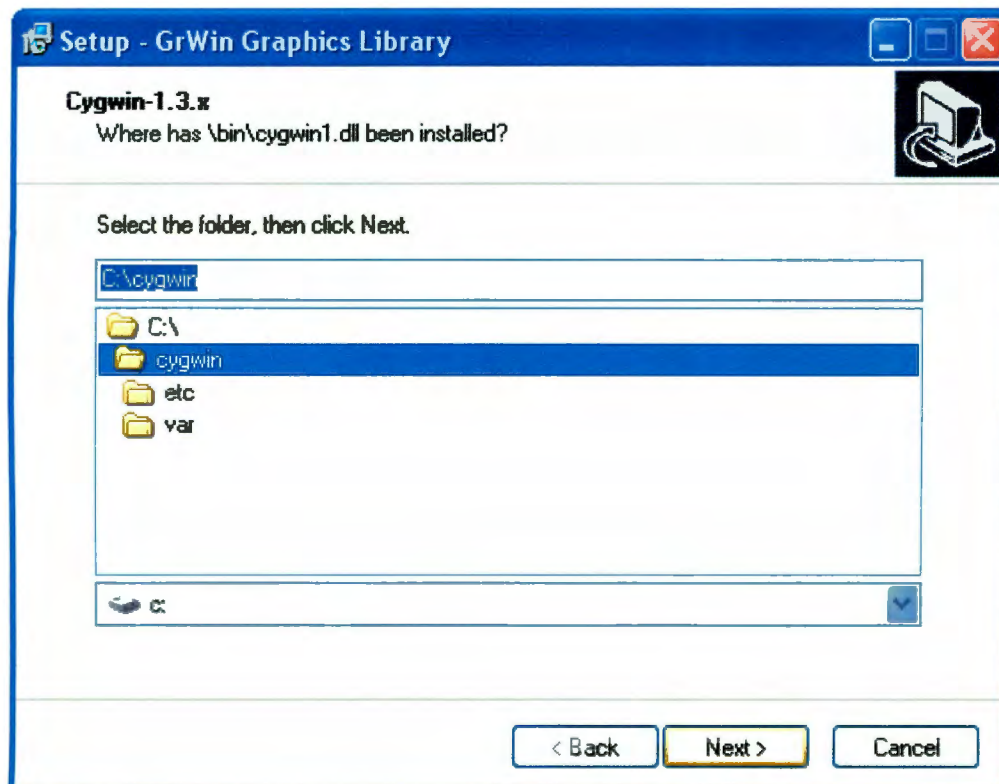
14. Seleccionar que es para Cygwin 1.3 v teclear "Next"



15. Seleccionar las tres opciones y teclear "Next"



16. Seleccionar la opción "Install"



17. Seleccionar el f3lder donde se instal3 Cygwin en este caso C:\Cygwin y teclear "Next"
18. Agregar al path de la computadora la linea C:\Cygwin\bin

D) INSTALACION DEL SOFTWARE DE LA TESIS

19. Del DISCO copiar el f3lder Tesis y colocarlo en C:\Cygwin\home\\
20. Verificar que en el f3lder C:\Cygwin\home\\Tesis se encuentren los siguientes archivos:

Evolucion.c
 BaldwinEffect.c
 Simulador.c
 TiempodeVidaEvolucion.txt
 TiempodeVidaAprendizaje.txt
 TiempodeVidaBaldwinEffect.txt
 Pesos.txt
 PesosBaldwinEffect.txt
 PesosEvolucionados.txt

E) COMPILACI3N DE LOS ARCHIVOS

En el directorio C:\Cygwin\home\\Tesis\

EVOLUCI3N

21. Teclear gcc -o Evolucion Evolucion.c

EFECTO BALDWIN

22. Teclear gcc -o BaldwinEffect BaldwinEffect.c

SIMULADOR

23. Teclear gcc -o Simulador Simulador.c -lgrwin -mwindows
24. Verificar que se hallan generado en el directorio C:\Cygwin\home\\Tesis\ los archivos :

Evolucion.exe
 BaldwinEffect.exe
 Simulador.exe

MANUAL DE OPERACIÓN

En el directorio C:\Cygwin\home\\Tesis\

EVOLUCIÓN

25. Teclear ./Evolución

En el archivo PesosEvolucionados.txt aparecerán los 16 resultado de la evolución
En el archivo TiempodeVida.txt aparecerá el Tiempo de vida del Sistema

EFECTO BALDWIN

26. Copiar los Pesos Evolucionados al archivo pesos mediante la directiva

```
cp PesosEvolucionados.txt Pesos.txt
```

27. Teclear ./BaldwinEffect

En el archivo PesosBaldwinEffect.txt aparecerán los 16 pesos resultado del Efecto Baldwin.

En el archivo TiempodeVida.txt aparecerá el Tiempo de vida y las Presas

SIMULADOR

28. Copiar los Pesos que se desean simular mediante la directiva

```
cp <archivo con los pesos que se quieren simular>.txt Pesos.txt
```

29. Teclear ./Simulador

En el archivo TiempodeVidaSimulador.txt aparecerá el tiempo de vida y las presas