



Universidad Nacional del Sur

Tesis de Magister en Computación Científica

**UN ENFOQUE DE SIMULACIÓN BASADA EN AGENTES EN
PROCESOS ESTRATÉGICOS DE FORMACIÓN DE REDES**

Juan M.C. Larrosa

Bahía Blanca

Argentina

2010

A Papá, Mamá y a Txopí, un recuerdo y una esperanza...

Agradecimientos

Esta tesis está dedicada a tres personas: mis padres Oscar Fernando y Herminia, quiénes se han ido tempranamente dejando un vacío muy difícil de llenar, y a mi hija Sofía Ainhoa, quien cumple su segundo año en este mundo. Ambos representan el pasado y futuro de amor para mi familia.

Deseo agradecer a mi familia, mis hermanos: Carlos, Oscar, Daniel y Martín.

A mis amigos del alma Carlos Scoccia y Gustavo Duprat por su compañía y amistad en todos estos tiempos.

A Ignacio Ponzoni, por su gran paciencia y empuje para terminar esta dilatada tesis de Magister.

A Fernando Tohmé, por su permanente consejo, amistad e intercambio de ideas. Fernando, por sí solo, representa el estado del conocimiento científico de la teoría económica en Bahía Blanca y me siento agradecido siempre haberlo conocido y que me haya brindado su amistad.

Tabla de contenidos

	página
Capítulo 1. Introducción	
1.1 Introducción	8
Capítulo 2. Juegos de formación de red	
2.1 Introducción	10
2.2 Sobre la formación de redes y sus soluciones analíticas	10
2.2.1 Juegos de formación de red	13
2.2.2 Definiciones	16
2.2.3 Las conexiones y los beneficios	17
2.2.4 Modelo de Bala y Goyal (2000)	20
2.2.4.1 Hipótesis	22
2.2.5 Modelo de Larrosa y Tohmé (2003b, 2005)	22
2.2.5.1 Hipótesis	23
2.3 Modelos de colaboración entre firmas	24
2.3.1 Estructura y equilibrio	25
2.3.1.1 Hipótesis	28
Capítulo 3. Agentes y algoritmos genéticos	
3.1 La modelación basada en agentes	29
3.2 Algoritmos genéticos como agentes de la economía	36
3.3 Los algoritmos genéticos	37
3.3.1 Esquema y definiciones	39
3.3.2 Codificación y población inicial	43
3.3.2.1 Representando redes como cadenas binarias	44
3.3.3 Función de performance	47
3.3.3.1 Beneficios directos	49
3.3.3.2 Beneficios indirectos	49
3.3.4 Los operadores	53
3.3.4.1 Operador de selección	53
3.3.4.2 Operador de cruzamiento	60
3.3.4.3 Operador de mutación	62
3.3.5 Principales parámetros	63
3.4 Los algoritmos genéticos y su interpretación económica	67
3.5 La estabilidad en el aprendizaje por algoritmos genéticos	70
Capítulo 4. Análisis de datos	
4.1 Introducción	73
4.2 Patrones de datos para cada hipótesis de investigación	74
4.2.1 Modelo de Bala y Goyal	75
4.2.1.1 Red de 5 nodos	76
4.2.1.2 Red de 10 nodos	82
4.2.1.3 Red de 20 nodos	88
4.2.1.4 Efecto del tamaño de la población	92
4.2.1.5 Efecto de la selección y mutación	95
4.2.1.6 Conclusiones parciales	96

4.2.2 Modelo de Larrosa y Tohmé	98
4.2.2.1 Red de 5 nodos	99
4.2.2.2 Red de 10 nodos	102
4.2.2.3 Red de 20 nodos	107
4.2.2.4 Efecto del tamaño de la población	111
4.2.2.5 Efecto de la selección y mutación	113
4.2.2.6 Conclusiones parciales	114
4.2.3 Modelo de Goyal y Joshi	116
4.2.3.1 Red de 15 nodos	117
4.2.3.2 Red de 30 nodos	121
4.2.3 Modelo de Billand y Bravard	122
4.2.3.1 Red de 10 nodos	122
4.2.3.2 Red de 20 nodos	124
4.2.4 Conclusiones parciales	126
4.3 Conclusiones	127
Capítulo 5. Conclusiones e implicaciones	
5.1 Introducción	128
5.2 Conclusiones acerca de las hipótesis de investigación	128
5.3 Conclusiones acerca de la problemática general de la investigación	129
5.4 Implicaciones para la teoría	129
5.5 Limitaciones	130
5.6 Futuros caminos de investigación	130
Apéndice 1. Glosario	131
Apéndice 2. Cálculo de distancias de recorrido de la información	132
Apéndice 3. Programación en Netlogo	134
Referencias	147

Tablas de Tablas, Figuras y Gráficos

Tabla 1. Pseudo-código para el Juego de Red de Comercio (Trade Network Game)	33
Tabla 2. Un mercader del juego TNG como agente de software	34
Tabla 3. Pseudo-algoritmo genético	41
Tabla 4. Correspondencia entre símbolos del modelo formal al computacional.....	41
Tabla 5. Enfoques usados en AG en la literatura económica	70
Tabla 6. Parámetros del AG.....	76
Tabla 7. Valores críticos para red de 5 nodos en caso monodireccional de Bala y Goyal (2000)	76
Tabla 8. Red de 5 nodos, comunicación monodireccional con decaimiento ($\delta = 0.5$).....	77
Tabla 9. Valores críticos para red de 5 nodos en caso bidireccional de Bala y Goyal (2000).....	79
Tabla 10. Red de 5 nodos, comunicación bidireccional con decaimiento ($\delta = 0.5$).....	80
Tabla 11. Valores críticos para red de 10 nodos en caso monodireccional de Bala y Goyal (2000).....	82
Tabla 12. Red de 10 nodos, comunicación monodireccional con decaimiento ($\delta = 0.5$).....	83
Tabla 13. Valores críticos para red de 10 nodos en caso bidireccional de Bala y Goyal (2000).....	85
Tabla 14. Red de 10 nodos, comunicación bidireccional con decaimiento ($\delta = 0.5$).....	86
Tabla 15. Valores críticos para red de 20 nodos en caso monodireccional de Bala y Goyal (2000).....	88
Tabla 16. Comunicación monodireccional, tasa de decaimiento 0.5	89
Tabla 17. Valores críticos para red de 20 nodos en caso bidireccional de Bala y Goyal (2000).....	91
Tabla 18. Comunicación bidireccional, tasa de decaimiento 0.5	91
Tabla 19. Tamaño de red y su efecto en ratio de acierto, valores medios y generaciones promedio .	93
Tabla 20. Ratio de acierto, Densidad de valores intermedios y generaciones promedio para comunicación monodireccional.....	94
Tabla 21. Ratio de acierto, densidad de valores intermedios y generaciones promedio para comunicación bidireccional.....	95
Tabla 22. Procedimiento de selección y su efecto en ratio de acierto y generaciones promedio	96
Tabla 23. Tasa de mutación y su efecto en ratio de acierto y generaciones promedio	96
Tabla 24. Parámetros del AG.....	98
Tabla 25. Comunicación monodireccional, 5 nodos	99
Tabla 26. Comunicación bidireccional, 5 nodos	101
Tabla 27. Comunicación monodireccional, 10 nodos	102
Tabla 28. Comunicación bidireccional, 10 nodos	104
Tabla 29. Comunicación monodireccional, 20 nodos	106
Tabla 30. Comunicación bidireccional, 20 nodos	109
Tabla 31. Tamaño de red y su efecto en ratio de acierto y generaciones promedio	111
Tabla 32. Ratio de acierto y generaciones promedio para comunicación monodireccional	112
Tabla 33. Ratio de acierto, Densidad de valores intermedios y generaciones promedio para comunicación bidireccional.....	113
Tabla 34. Procedimiento de selección y su efecto en ratio de acierto y generaciones promedio para comunicación bidireccional.....	114
Tabla 35. Tasa de mutación y su efecto en ratio de acierto y generaciones promedio para comunicación bidireccional.....	114
Tabla 36. Parámetros del AG.....	116
Tabla 37. Resultados de las simulaciones para un φ bajo (15 nodos)	117
Tabla 38. Resultados de las simulaciones para un φ alto (15 nodos).....	119
Tabla 39. Resultados de las simulaciones para un φ bajo (30 nodos)	121
Tabla 40. Parámetros del AG.....	122
Tabla 41. Resultados de las simulaciones, 10 nodos	123
Tabla 42. Resultados de las simulaciones, 20 nodos	125
Figura 1. Arquitectura de redes.....	17

Figura 2. Esquema de conexiones totales	19
Figura 3. Modelo de segregación de Schelling (1978)	32
Figura 4. Ejemplo 1	46
Figura 5. Estabilidad de densidad de red y beneficios en una red de 10 nodos	72
Figura 6. Estabilidad de densidad de red y beneficios en una red de 30 nodos	72

Gráfico 1. Beneficio promedio final y costo relativo de la información (5 nodos).....	78
Gráfico 2. Densidad de red final y costo relativo de la información (5 nodos)	78
Gráfico 3. Densidad de red final y generaciones (5 nodos)	79
Gráfico 4. Beneficio promedio final y costo relativo de la información (5 nodos).....	81
Gráfico 5. Densidad de red final y costo relativo de la información (5 nodos)	81
Gráfico 6. Densidad de red final y costo relativo de la información (5 nodos)	82
Gráfico 7. Beneficio promedio final y costo relativo de la información (10 nodos).....	84
Gráfico 8. Densidad de red final y costo relativo de la información (10 nodos)	84
Gráfico 9. Densidad de red promedio y ratio de aciertos (10 nodos).....	85
Gráfico 10. Beneficio promedio y costo relativo de la información (10 nodos).....	87
Gráfico 11. Densidad de red final y costo relativo de la información (10 nodos)	87
Gráfico 12. Densidad de red y generaciones promedio (10 nodos)	88
Gráfico 13. Costo relativo de la información y beneficio promedio (20 nodos)	90
Gráfico 14. Densidad de red y costo relativo de la información (20 nodos).....	90
Gráfico 15. Costo relativo de la información y beneficio promedio (20 nodos)	92
Gráfico 16. Costo relativo de la información y densidad promedio (20 nodos).....	92
Gráfico 17. Beneficio promedio y costo relativo de la información (5 nodos).....	99
Gráfico 18. Densidad promedio y costo relativo de la información (5 nodos).....	100
Gráfico 19. Beneficio promedio y generaciones (5 nodos)	100
Gráfico 20. Beneficio promedio y valor relativo de la información (5 nodos)	101
Gráfico 21. Densidad promedio y valor relativo de la información (5 nodos)	102
Gráfico 22. Beneficio promedio y costo relativo de la información (10 nodos).....	103
Gráfico 23. Densidad promedio y costo relativo de la información (10 nodos).....	103
Gráfico 24. Densidad promedio y generaciones (10 nodos)	104
Gráfico 25. Beneficio promedio y costo relativo de la información (10 nodos).....	105
Gráfico 26. Densidad promedio y costo relativo de la información (10 nodos).....	105
Gráfico 27. Densidad promedio y generaciones (10 nodos)	106
Gráfico 28. Beneficio promedio y costo relativo de la información (20 nodos).....	107
Gráfico 29. Densidad de red y costo relativo de información (20 nodos)	108
Gráfico 30. Densidad de red y tasa de mutación (20 nodos)	108
Gráfico 31. Beneficio promedio y costo relativo de la información (20 nodos).....	109
Gráfico 32. Densidad promedio y costo relativo de la información (20 nodos).....	110
Gráfico 33. Densidad promedio y tasa de mutación (20 nodos)	111
Gráfico 34. Beneficio promedio y tasa de mutación (15 nodos)	117
Gráfico 35. Densidad de red promedio y tasa de mutación (15 nodos).....	118
Gráfico 36. Densidad de red promedio y generaciones (15 nodos)	118
Gráfico 37. Beneficio promedio y tasa de mutación (15 nodos)	119
Gráfico 38. Densidad promedio y tasa de mutación (15 nodos)	120
Gráfico 39. Densidad promedio y generaciones (15 nodos)	120
Gráfico 40. Beneficios promedio y tasa de mutación (10 nodos).....	123
Gráfico 41. Beneficio promedio y generaciones (10 nodos)	124
Gráfico 42. Densidad de red promedio y generaciones (10 nodos)	124
Gráfico 43. Beneficios promedio y tasa de mutación (20 nodos).....	125
Gráfico 44. Densidad promedio y tasa de mutación (20 nodos)	126
Gráfico 45. Interface	136

Capítulo 1. Introducción

1.1 Introducción

La modelación individual de la interacción entre agentes en Economía ha recibido particular atención en los últimos años. Estos aportes han recaído inicialmente, y siguiendo la tradición académica, en modelos analíticos que contemplan la aparición de interacciones locales entre los individuos. Esta interacción permite, en algún punto, determinar resultados de conducta diferentes comparados con un análisis en el que los agentes se comporten aisladamente. La modelación de dicha interacción en términos formales ha utilizado herramientas matemáticas que permiten la traslación de efectos por parte de un individuo hacia otro. La teoría de juegos, dado que analiza en el comportamiento óptimo de un agente que se ve restringido en su accionar por las estrategias de otros agentes, resulta un modelo analítico clásico de ciertos casos de interacción. Otros métodos formales más recientes, los cuales intentan capturar formas más generales de interrelación han visto sustentada su investigación en herramientas provenientes de la matemática (teoría de grafos, grillas, entre otras) así como de la física (teoría del campo medio). En este sentido los aportes de Ioannides (2001) y Blume y

Durlauf (2000) resultan excelentes síntesis de este enfoque analítico de modelación de la interacción de agentes.

El enfoque basado en agentes es otra forma de modelar esta interacción. Sus resultados no son formales sino numéricos. Su sustento no es básicamente matemático sino computacional. Los agentes resultan ser bajo este marco de investigación objetos de programas. Los objetos agente tienen estados y reglas de comportamiento. Correr un modelo de agentes comprende la inicialización de la población de agentes, el proceso subsiguiente de interacción entre los mismos y el monitoreo del resultado final. En definitiva, correr el modelo en el tiempo la suficiente cantidad de veces, en caso de ser estocástico, es lo único que se necesita para 'resolverlo'.

Un primer objetivo general de esta tesis es profundizar en las capacidades de la simulación basada en agentes como herramienta para resolver o complementar la resolución de modelos y casos formales de la Economía, en este caso específico, la eficiencia en los procesos estratégicos de formación de redes. Más precisamente, el propósito particular del presente trabajo es introducir el análisis evolutivo computacional desde la perspectiva del aprendizaje, entendiendo a éste como un proceso de análisis, evaluación de información y corrección de estrategias en juegos repetidos de formación de redes. El algoritmo genético, en ese sentido, representa una estructura natural de análisis. El proceso de aprendizaje se aprecia directamente observando la convergencia de los valores de las variables relevantes hacia su estado estacionario coincidente (o cercano) con el valor predicho por la teoría económica. El

trabajo experimental encuentra grados de convergencia (aprendizaje) alentadores, para un código desarrollado para tal efecto.

La tesis se compone del Capítulo 2, en el cual se realizará una breve revisión de los modelos teóricos de formación de redes y las hipótesis que se desprenden de los mismos. El Capítulo 3 repasará brevemente el estado de la economía computacional basada en agentes, para luego introducirse específicamente en los algoritmos genéticos, su modelización y aplicación y su interpretación económica. El Capítulo 4 analizará los resultados obtenidos en los trabajos de simulación realizados y finalmente, el Capítulo 5 concluirá con discusiones respecto a los resultados y alcance de los hallazgos.

Capítulo 2. Juegos de formación de red

2.1 Introducción

El análisis de la literatura teórica empezará con una revisión de los trabajos principales que inspiraron las hipótesis de investigación de la presente tesis. Se dará inicio entonces al repaso de trabajos que presentan un esquema inicialmente genérico de formación de red en la que simplemente se comparan ingresos y costos de la transmisión de información a través de distintas topologías y protocolos de intercambio. Seguidamente se explorará brevemente un par de modelos adicionales pertenecientes a la economía industrial en los cuales las decisiones de creación de redes actúan como reductoras de costos de producción.

2.2 Sobre la formación de redes y sus soluciones analíticas

Para el caso que atañe a esta investigación debemos considerar un tema de reciente interés en la teoría económica: los juegos de formación de redes estudiados bajo la conducta estratégica de los agentes. Bala y Goyal (2000) es, para el caso de formación de redes

basadas en teoría de juegos no cooperativos, el aporte inicial. El modelo representa a la formación de redes sociales entre agentes como un juego de asociación óptima por el cual los participantes analizan costos de conexión con los beneficios de información por acceder a nuevas fuentes de información o nodos de la red. Como resultado del análisis se observan topologías particulares en la estructura óptima de red según varíe la dirección de conexión entre los agentes (comunicación mono o bidireccional) así como la estructura de beneficios subyacente. En este modelo los agentes son homogéneos y poseen esquemas de información completa (conocen el modelo y sus limitaciones, conocen a sus pares y sus estrategias de conducta por lo que predicen con exactitud sus futuros movimientos, aunque en este modelo en particular los agentes son miopes, es decir actúan según lo observado en jugadas anteriores recientes sin recordar toda la historia del juego, entre otras características). En general los modelos de formación de redes se asientan en supuestos simples: (1) La formación de enlaces es un proceso binario de mutuo consentimiento, (2) La formación de enlaces es costosa, (3) La clase de la función de pago debiera ser lo mas general posible. Para el caso de Bala y Goyal (2000) el modelo determina que, por ejemplo, en condiciones de comunicación monodireccional y bajo beneficios estrictamente positivos de enlace la estructura de conexión óptima para los agentes es la red circular o ciclo.

Esta particular topología es hallada para otros modelos de formación de redes, tales como aquellos de juegos secuenciales en donde el supuesto de miopía es relajado manteniéndose altos costos iniciales de conexión, por ejemplo Watts (2002). Otro caso interesante y reciente es el planteado por Gilles y Sarangi (2003) quienes emplean un modelo de formación de redes pero aquí los agentes establecen expectativas a priori acerca de la conducta de sus pares. En esta modelización no se considera información perfecta por parte

de los agentes. Con un periodo de anticipación los agentes los agentes establecen probabilidades positivas de que los otros agentes aceptarán su propuesta de formas enlaces siempre que ello genere algún beneficio positivo para ambos. Esta 'confianza' depositada entre los agentes ayuda a coordinar las acciones en este juego de información incompleta (la estabilidad de estos juegos es denominada monádica). La modelación de esta variable de origen marcadamente sociológico mediante el uso de expectativas de los agentes sobre los movimientos de sus pares genera, en caso de coincidencia entre los valores esperados y los realmente observados, equilibrios estables (redes estables con topologías diferenciables) .

La formación de redes presentada por Bala y Goyal (2000) para el caso particular de comunicación monodireccional presenta un resultado de topología de red coincidente en muchos casos, aunque no en todos, al planteado por los aportes de Larrosa y Tohmé (2003b, 2005). Las diferencias principales entre ambas líneas de análisis residen en las diferentes formas de valorar los costos de conexión de ambos trabajos¹. Esto se discutirá a continuación.

2.2.1 Juegos de formación de red

Los modelos iniciales que consideraremos analizan a las redes como grafos dirigidos con flujos de información en un solo sentido o monodireccionales. Llamamos 'información' (en un uso más bien genérico del término) al bien de utilidad que fluye en las redes. Cada agente está dotado con alguna cantidad de información, pero tiene una función de pagos que depende positivamente del monto de información a la cual ella tiene acceso. Estableciendo enlaces con otros agentes ella puede tener la información poseída por estos pero ella tiene que compartir el pago del gasto de conexión de esta información a la que ella accede. Esta

¹ Estas diferencias se tratan muy bien en Larrosa y Tohmé (2003a)

es una forma de representar el hecho de que en muchos casos la información obtenida indirectamente requiere, de todos modos, una cierta cantidad de colaboración con la fuente de modo de incentivarla a seguir enviando la información en el futuro. El enfoque de costeo compartido aplicado en este trabajo supone que cada agente paga una pequeña tarifa a cada enlace en el sendero que le permite alcanzar la información deseada. Aquí, los beneficios y costos asociados son simétricos. Quién solventa el costo de conexión es quién recibe el beneficio de ello². El problema es determinar cuál estructura puede emerger como equilibrio estratégico entre los agentes y si ésta es óptima o no. Encontramos que los equilibrios estrictos de Nash o equilibrios de Nash con el mínimo número de enlaces sostienen una red circular, la cual es estable y óptima.

Las circunstancias del problema de formación de redes estudiadas aquí pueden ser observadas en muchos campos. Así, por ejemplo, considérese el siguiente escenario: supongamos que los usuarios de Internet son tarifados por un pequeño monto por cada enlace que ellos visitan. Cuando un usuario visita un sitio web, él tiene que pagar dicha tarifa. Si él sigue, a su vez, otro link tiene que pagar por la nueva conexión pero accede a mayor información. Si a su vez él sigue otro enlace más en este sitio él accede al nuevo sitio pagando de nuevo la tarifa pero accediendo a mayor información. La pregunta es: ¿cuál es la manera más eficiente de navegar los sitios bajo esta estructura de costos?

Más cercano a nuestro marco de estudio, podríamos preguntar qué clase de arquitectura para una red de área local (LAN – Local Area Network) incrementa la velocidad de flujo mientras al mismo tiempo reduce las pérdidas. Este es en efecto análogo a nuestro

² El tema de la simetría y asimetría de pagos en el modelo de Bala y Goyal (2000) está muy bien estudiado en Billand y Bravard (2005).

problema genérico: una computadora particular en la LAN podría necesitar recurrir a los recursos de otra computadora en la red. Debiera existir un protocolo eficiente para elegir a qué máquina conectarse. Al mismo tiempo tiene que pagar una pequeña 'tarifa' (en términos de tiempo de procesamiento, por ejemplo) para alcanzar la máquina que le proporciona la mayor cantidad de recursos. Dado que esto es cierto para todas las máquinas en la red, el resultado estratégico tiene que permitir a todos alcanzar la mayor cantidad de información disponible, pagando lo menos posible. El resultado final podría ser una red circular como en el caso de la arquitectura *Token Ring* de IBM (Tanenbaum, 1989).

También puede ocurrir esto en organizaciones humanas, por ejemplo los comités multidisciplinarios de evaluación. Están usualmente constituidos por expertos en diferentes campos. Cada uno debe confiar en el otro para obtener información de su campo en el cual es experto. El resultado final de la evaluación debe contemplar la evaluación de todos los miembros. La red circular, en este caso, minimiza el número de preguntas mientras al mismo tiempo maximiza la información disponible para todos.

Finalmente, podemos considerar la cuestión de los costos de conexión. De manera muy directa e intuitiva podemos pensar que los costos de conexión son directamente proporcionales a la distancia de conexión. Por ejemplo, si debo acceder a la información de una agente que se encuentra a un enlace de distancia simplemente pagaría el costo de conexión directo. Sin embargo, se puede pensar un ejemplo más complejo de una red cuya expansión sea altamente costosa. Es decir, acceder a través de esa red a un agente i que se encuentra a una distancia geodésica de 1 cuesta c si el agente j no se ha conectado con nadie más pero si este ha establecido 4 conexiones, i debe pagar $4c$ por acceder a la

información de j y su entorno. Siendo el acceso a esta red tan acumulativamente costoso (por un ejemplo, una red de informantes quiénes cobran por su información por todos los agentes previos a quiénes ellos debieron consultar) debe plantear una optimización especial. Ahora continuamos con las definiciones del modelo.

2.2.2 Definiciones

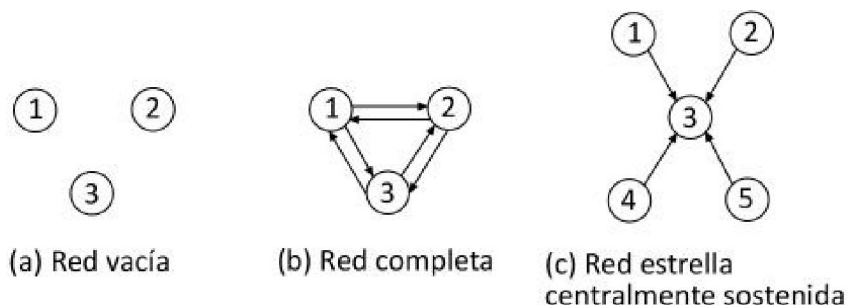
A continuación se definen conceptos y herramientas comunes al desarrollo de toda la argumentación del presente trabajo. Sea $N = (1, \dots, n)$ un conjunto de agentes. Para evitar resultados triviales siempre vamos a suponer que $n \geq 3$. Si i y j son dos miembros típicos de N , un enlace entre ellos, sin intermediarios, originado en i y finalizado en j será representado como ij . La interpretación de ij es que i establece un contacto con j que permite a i tener acceso a la información de j así como a su red de contactos.

Como fue mencionado, una red puede ser vista como un grafo. Una decisión importante de modelización es la de sí el grafo va a ser *dirigido* o *no dirigido*. Esta elección de primitivas tiene consecuencias también para los resultados de equilibrios en juegos no cooperativos de formación de redes. Cada agente $i \in N$ tiene alguna información de su propiedad, $v_i \in \mathbb{R}_+$ (es decir, representado como un real positivo). Como se mencionó i puede acceder a más información formando enlaces con otros agentes. Los agentes tratarán de maximizar la utilidad de la información disponible para ellos así como minimizar el costo de conexión con otros agentes. Para lograr ello, ellos estarán dotados con un conjunto de estrategias. Cada estrategia para $i \in N$ es un vector $g_i = \langle g_{i,1}, \dots, g_{i,i-1}, g_{i,i+1}, \dots, g_{i,n} \rangle$ de $(n - 1)$ dimensiones donde cada $g_{i,j}$ para $j \neq i$ vale 0 o 1. Esto es interpretado como significando que i establece un enlace directo con j si $g_{i,j} = 1$ mientras que si $g_{i,j} = 0$ dicho enlace no existe. El conjunto

de todas las estrategias se anota como G_i . Restringimos nuestro análisis sólo a los casos de estrategias puras, lo cual implica que $G_i = 2^{n-1}$. Finalmente, $G = G_1 \times \dots \times G_n$ denota el conjunto de perfiles de estrategia en la interacción entre los agentes de N .

A continuación definimos las principales configuraciones de redes que son usadas extensivamente en nuestro trabajo. Una red g es *completa* si para cada par de agentes i y j , existe un enlace desde i hacia j . La red completa se anota como g^c . Una red g es una *estrella centralmente sostenida* si y sólo si existe un agente o firma i tal que i ha formado un enlace con cada firma j , y cada $j \neq i$ no ha formado ningún enlace (ver Figura 1). La estrella apuntada hacia adentro es anotada como g^s . Una red g es *vacía* si no hay ningún agente que haya formado ningún enlace. Esta red es denotada como g^e .

Figura 1. Arquitectura de redes



Pasemos a continuación a analizar cómo se contabilizan los costos de conexión y los beneficios de la información recibida.

2.2.3 Las conexiones y los beneficios

Para nuestro análisis en particular será fundamental modelar las formas de conexión que prevalecerán en la determinación futura de las funciones de pagos y de costos de cada

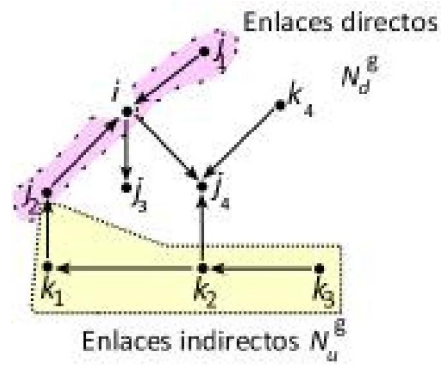
agente. En nuestro modelo, será también importante distinguir los enlaces directos del agente de sus enlaces indirectos. Los enlaces directos son aquellos enlaces que tienen un sendero de valor uno mientras que los enlaces indirectos son aquellos que tienen un sendero de valor superior a uno, como ya se definió en la sección previamente.

$$N^{i;g} = \{k \in N | g_{i,k} = 1\} \cup \{h \neq k | h \in N | g_{i,h} \geq 2\} \quad (2.1)$$

Nuevamente, $N^{i;g}$ es un vector que recolecta todos aquellos agentes que directa o indirectamente están conectados con i . Aquí también queda discriminado el grado de cercanía de los enlaces. El primer término de (2.1) se corresponde a los *enlaces directos* del agente i mientras que el segundo término refiere a los *enlaces indirectos* del agente, los cuales ya veremos también tiene que solventar. Esto es, los pagos de i son la suma de toda la información que puede ser accedida por ella, menos el costo de los senderos (es decir conexiones directas e indirectas, véase la Figura 2 para una ilustración al respecto con grafos dirigidos con flujo monodireccional) que llegan a ella que están establecidos de acuerdo a g (recuérdese que cada enlace se supone tiene costo unitario). La intuición aquí es que i obtiene un pago por acceder a mayor información pero al mismo tiempo ella tiene que pagar un cargo o arancel por cada uno de los enlaces en los senderos hacia las fuentes de información. Esta forma de financiar las conexiones es, por lo tanto, acumulativa. A medida que la información pasa por más agentes se hace más cara, dado que hay que solventar a todos los intermediarios. En la Figura 2 se observa como i se conecta directamente con j_1 y j_2 , de los que obtiene información directamente (conjunto de puntos espaciados). Pero también obtiene información indirecta dado que j_2 está conectado con k_1 , siendo también que k_1 se conectó con k_2 y ésta con k_3 (conjunto de puntos densos). En definitiva, i accede a

la información de los dos conjuntos y debe pagar por los enlaces incluidos dentro de ambos conjuntos. Obsérvese que j_3 , j_4 y k_4 nunca son accedidos por el agente i .

Figura 2. Esquema de conexiones totales



Vamos a formalizar ahora la *función de pagos* de cada agente. Ella posee una información propia, v_i , pero puede acceder a poseer más información si crea una función de conexiones y accede a las informaciones propias y obtenidas por los otros agentes. Por ello, formalizando el valor de la información total accedida por i , I_i , nos queda:

$$I_i(g) = v_i + \sum_{j \neq i | j \in g} I_j(N^{i:g}), \quad (2.2)$$

donde I_j es el valor de la información del agente j que es accedido por i a través del perfil de estrategias g la que se representa por $N^{i:g}$. Ello determina la estructura de conexión del agente i y la información diversificada a la que accede.

Para hacer a este esquema un juego tenemos que definir los beneficios de los agentes.

Supondremos $\Pi_i : G \rightarrow R$, la función de beneficios para el agente i , que será:

$$\Pi_i(g) \equiv I_i(g) - c \sum_{j \in N^{i;g}} \mu_j(g) \quad (2.3)$$

donde I_i representa la información poseída y accedida por el agente i según la estrategia de conexiones g de la definición (2.1), c representa el costo de cada conexión y $\mu_j(g)$, tal como se definió en la sección anterior, la cantidad de enlaces directos e indirectos que debe solventar la agente i . Con fines de recalcar algún aspecto particular de una estrategia de conexión, no sin un dejo de abuso notacional, se describirá en algunas ocasiones a $I_i(g)$ como $I_i(N^{i;g})$ y al conjunto de información de algún agente j accedido por i como $I_j(N^{i;g})$.

En las redes reales, sin embargo, la información al pasar de nodo en nodo puede perder calidad, tanto sea porque tarda tiempo en traspasar un nodo al otro como por la adición de ruido en cada traspaso. Ello hace que la información pierda valor a medida que sea más largo el trayecto en términos de nodos intermediarios entre el origen y el destino final de la misma. En nuestro contexto la pérdida de calidad de la información ó decaimiento (decay) va a ser representada por un parámetro $\delta \in [0,1]$ donde $\delta = 0$ implica decaimiento completo de la información mientras que $\delta = 1$ implica ausencia de pérdida de calidad de la información. Si definimos a $d(i,j;g)$ como el sendero de recorrido más corto (shortest path) entre i y j en la red g . Entonces (2.3) queda representado como:

$$\Pi_i(g) = 1 + \sum_{j \in N_i(g)} \delta^{d(i,j;g)} - c \sum_{j \in N^{i;g}} \mu_j(g) \quad (2.4)$$

Un último caso por analizar es aquel en el que podemos suponer que los costos se hacen crecientemente más alto a medida que nos alejamos de la fuente de información. Se puede

suponer que dicho incremento en el costo es directamente proporcional a la distancia entre los agentes. De este modo, (2.3) quedaría de la siguiente forma:

$$\Pi_i(g) = I_i(g) - c \sum_{j \in N^i(g)} d(i, j; g) \mu_j(g) \quad (2.5)$$

Pasemos ahora a realizar una muy concisa revisión de las contribuciones de juegos de formación red que se utilizarán como referencia del modelo computacional.

2.2.4 Modelo Bala y Goyal (2000)

El modelo de Bala y Goyal (2000) se caracteriza por su simpleza. Los agentes forman enlaces directos con el fin de intercambiar información comparando costos y beneficios de dicha operación. El modelo analiza el caso de comunicación unidireccional entre los agentes (grafo dirigido) y analiza el caso de comunicación bidireccional (grafo no dirigido). Las estructuras de equilibrio resultan para el primer caso la red vacía cuando los costos de conexión superan el número de agentes menos uno y la red circular en el caso en el caso en que el costo de conexión se sitúe entre cero y uno.

La función de beneficios de los agentes viene dada por el resultado neto del valor de la información accedida directa menos los costos de conexión directos. Es decir, no consideramos ni la información obtenida indirectamente ni costos de conexión indirectos.

Los resultados que pretendemos contrastar con un modelo de computación evolutiva son los siguientes:

2.2.4.1 HIPÓTESIS

Tanto para el modelo de comunicación directa monodireccional como para el modelo de comunicación directa bidireccional se postula la siguiente proposición:

- a) *Si la estructura de pagos viene dada por (2.3), entonces una red de Nash es vacía o mínimamente conectada. (Proposición 3.1 y Proposición 4.1, BG2000)*

Para el modelo de comunicación monodireccional directa con tasa de decaimiento en la información δ (BG2000: 1194-1202)

- i. *Dada la estructura de pagos (2.3), una red con comunicación monodireccional y presencia de pérdida de calidad de información tiene una estructura eficiente si :*
- a. *Es una red completa cuando $0 < c < 2[\delta - \delta^2]$ (H1),*
 - b. *Es una red de estrella cuando $2[\delta - \delta^2] < c < 2\delta + (n-2)\delta^2$ (H2), y*
 - c. *Es una red vacía cuando $c > 2\delta + (n-2)\delta^2$ (H3) (Proposición 4.1 , BB2000)*

Para el modelo de comunicación bidireccional directa (BG2000: 1202-1209)

- ii. *Dada la estructura de pagos (2.3), una red con comunicación bidireccional y presencia de pérdida de calidad de información tiene una estructura eficiente si :*
- a. *Es una red completa cuando $c < \delta - \delta^2$ (H4),*
 - b. *Es una red de estrella cuando $\delta - \delta^2 < c < \delta + [(n-2)/2]\delta^2$ (H5), y*
 - c. *Es una red vacía cuando $c > \delta + [(n-2)/2]\delta^2$ (H6) (Proposición 4.1 , BB2000)*

Repasemos a continuación un modelo de formación de redes donde los costos se acumulan a medida que se hace más largo el sendero entre el origen y el destino de la información.

2.2.5 Modelos de Larrosa y Tohmé (2003b, 2005)

Estos modelos son adaptaciones del modelo de Bala y Goyal para el caso de comunicación unidireccional (Larrosa y Tohmé, 2003b) y bidireccional (Larrosa y Tohmé, 2005) cuando la estructura de costos de conexión es acumulativa en la cantidad de nodos que sean requeridos para obtener dicha información. En el primer caso los resultados son similares a

los obtenidos por los autores indios (es decir en caso de costos de conexión muy altos la red de equilibrio hacía metas que si los costos de conexión son muy bajos la red de equilibrio es la red circular) mientras que en el segundo caso los resultados difieren. El caso de comunicación bidireccional con costos de conexión acumulativos la red de equilibrio no trivial es la red de línea (con nodos intermedios de conexión).

Especialmente para el modelo de comunicación indirecta monodireccional también como para el modelo de comunicación indirecta bidireccional.

2.2.5.1 HIPÓTESIS

Para el modelo de comunicación monodireccional directa e indirecta con costos acumulativos de conexión (Larrosa y Tohmé, 2003b) sin decaimiento se plantea las siguientes hipótesis:

- III. *Dada la estructura de pagos (2.5), una red de comunicación monodireccional eficiente es vacía o es circular. Específicamente si costo comparado con el pago c es menor que 1 para cada uno de los miembros ($c < 1$), entonces la red circular (ciclo) es la única red eficiente (H7). Finalmente, si el costo es mayor que 1 ($c > 1$) entonces la red vacía es la única red eficiente (H8).*

Finalmente, para el modelo de comunicación bidireccional directa e indirecta con costos acumulativos de conexión (Larrosa y Tohmé, 2005) sin decaimiento nos plantea las siguientes hipótesis:

- IV. *Dada la estructura de pagos (2.5), una red de comunicación bidireccional eficiente es vacía o es una red de línea. Específicamente si costo c es positivo pero menor que 1 ($0 < c < 1$) entonces la red de línea es red eficiente (H9). Si el costo comparado con el pago es mayor 1 ($c > 1$) la red vacía es la única red eficiente (H10).*

Veamos ahora un breve repaso a los modelos de formación de redes para establecer colaboración o espionaje entre firmas.

2.3 Modelo de colaboración entre firmas

Los modelos de Goyal y Joshi (2003) y Billand y Bravard (2004) son modelos en los que se analiza la interconexión de empresas para realizar intercambio de información bidireccional y monodireccional respectivamente. Las redes aquí representan el beneficio de la externalidad de conocer información (técnica y tecnológica, legal, comercial, publicitaria, entre otras) que permita a la firma que inicia y sostiene el enlace reducir costos de producción. Conectarse entonces reduce los costos. Cuando el flujo de información es simétrico (grafos no dirigidos; Goyal y Joshi, 2003) ambas firmas inician un enlace mutuamente beneficioso. Usualmente es el caso de los joint-ventures o acuerdos de colaboración entre firmas. Cuando el flujo de información es asimétrico (grafos dirigidos; Billand y Bravard, 2004) la firma que inicia el enlace accede a información de la firma enlazada y esta no puede rechazar dicho pedido de información. Situaciones que pueden ser comprendidas por esta modelización son, por ejemplo:

- (i) Las firmas visitan a sitios web de sus firmas competidoras para observar su oferta de bienes y servicios, precios, personal empleado, e imagen de mercado,
- (ii) La consulta de balances públicos en el caso de firmas que coticen en Bolsa de Valores a fin de evaluar su performance económica,
- (iii) La consulta del registro de patentes para observar las patentes obtenidas por los competidores,
- (iv) La lectura de revistas especializadas sobre las prácticas y formas de operar de la competencia,
- (v) Formas ilegales de adquisición de información, como espionaje industrial, entre otras.

A continuación definimos conceptos necesarios para la obtención de los resultados y sus alcances y limitaciones. Siguiendo con el esquema del modelo, se describe la función de costos específica del modelo.

2.3.1 Estructura y equilibrio

Las firmas primero eligen sus enlaces, luego observan la red resultante y compiten en el mercado. Entonces, hay dos etapas en el juego. Más precisamente, en la etapa I, cada firma decide cual enlace establecer con otras firmas. La determinación de un enlace es una inversión en búsqueda de externalidades acerca de tecnología, productos o mercados. Las externalidades obtenidas pueden ser interpretadas como flujos de externalidades. Suponemos inicialmente que estos flujos no son simétricos cuando una firma establece un enlace con otra firma, la primera se beneficia de este enlace pero la inversa no es cierta.

Supondremos que establecer un enlace requiere de un costo de inversión fija dado por $\varphi > 0$. Suponemos que las firmas son inicialmente simétricas con costo fijo cero e idénticas funciones de costo. Consideramos que establecer un enlace es una forma de reducir costos. Más específicamente, suponemos que la función de costo marginal y la función de costo variable medio de una firma $i \in N$ tiene la misma forma funcional:

$$c_i(n_i(\mathbf{g})) = \gamma_0 - \gamma n_i(\mathbf{g}), \quad (2.3.1)$$

donde $\gamma_0, \gamma \in R_+^*$ tal que $\gamma_0 > \gamma(n-1)$. Una red \mathbf{g} induce un vector de costo variable medio para las firmas que esta dado por la siguiente función

$$c(n_i(\mathbf{g})) = \{c_1(n_i(\mathbf{g})), c_2(n_i(\mathbf{g})), \dots, c_i(n_i(\mathbf{g})), \dots, c_n(n_i(\mathbf{g}))\}$$

Los resultados en la etapa I, v.g. el conjunto de enlaces formados por las firmas, son perfectamente observables entre los competidores cuando las firmas toman decisiones acerca de las cantidades para producir en la etapa II. Dados estos enlaces, en la etapa II, las firmas eligen las cantidades que producen simultáneamente.

Sea q_i la cantidad producida por la firma i y p el precio de mercado. Modelamos un mercado con productos homogéneos y competencia en cantidades. Más específicamente, y a fin de ser simples, suponemos que la función de demanda inversa en el mercado de productos es lineal:

$$p = \begin{cases} \alpha - \sum_{i \in N} q_i & \text{si } \sum_{i \in N} q_i < \alpha \\ 0 & \text{si } \sum_{i \in N} q_i \geq \alpha \end{cases}, \text{ con } \alpha > 0 \quad (2.3.2)$$

A continuación se describen los conceptos de equilibrio en redes que serán fundamentales para comprender las topologías óptimas encontradas.

Una red $g \in G$ se dice un equilibrio si, dejando constante el conjunto de enlaces formados por las otras firmas, cualquier firma que es conectada hacia otra en $g \in G$ tiene un incentivo en mantener ese enlace. Más aún, cualquier firma que no está conectada hacia otra firma en $g \in G$ no tiene ningún incentivo en formar un enlace con esta firma.

Sea g' una red donde i es la única firma que no tiene los mismos links que en g . Definamos a $\Pi_i(n_i(g), n_{-i}(g))$ como los beneficios netos de las firmas $i \in N$. Una red g es una red de equilibrio, si para todo i , tenemos:

$$\Pi_i(n_i(g), n_{-i}(g)) \geq \Pi_i(n_i(g'), n_{-i}(g')), \quad \forall g' \in G \quad (2.3.3)$$

El modelo BB refleja dos topologías principales dependiendo del tipo de estructura de mercado. En la competencia de mercado de Cournot los resultados obtenidos son los siguientes:

$$q_i^*(n_i(g), n_{-i}(g)) = \frac{(\alpha - \gamma_0) + n\gamma n_i(g) - \gamma \sum_{j \neq i} n_j(g)}{n+1}$$

donde

$$(\alpha - \gamma_0) - (n-1)^2 \gamma > 0 \quad (2.3.4)$$

a fin de asegurar que se produzca una cantidad estrictamente positiva en cada firma.

En equilibrio, la cantidad producida por la firma i depende de sus propios enlaces y de los enlaces de las otras firmas actualmente en el mercado. El beneficio de equilibrio, llamado $\Pi_i(n_i(g), n_{-i}(g))$, es para todo $i \in N$,

$$\Pi_i(n_i(g), n_{-i}(g)) = (q_i^*(n_i(g), n_{-i}(g)))^2 - \varphi n_i(g) \quad (2.3.5)$$

Donde φ representa la inversión necesaria de cada firma para iniciar y mantener la estructura de enlaces creada. Esta variable es trascendental porque ayuda a definir umbrales topológicos para las redes de conexión. Ahora caracterizamos la eficiencia en redes bajo competencia en cantidades.

2.3.1.1 HIPÓTESIS

- V. *Existiendo comunicación bidireccional, una red g es una red eficiente si $n_i(g) \in \{0, n-1\}$ para todo $i \in N$. Más aún,*
- si el costo de realizar un enlace, φ es bajo entonces g_c es la única red eficiente (H11);*
 - si el costo de realizar un enlace, φ es alto entonces g_e es la única red eficiente (H12) (Goyal & Joshi, 2003: 61)*

Prueba: Ver Goyal y Joshi (2003: 23).

- VI. *Existiendo comunicación unidireccional, una red g es una red eficiente si $n_i(g) \in \{0, n-1\}$ para todo $i \in N$. Más aún,*
- si el costo de realizar un enlace, φ es bajo entonces g_c es la única red eficiente (H13);*
 - si el costo de realizar un enlace, φ es alto entonces g_e es la única red eficiente (H14) (Billand & Bravard, 2004: 600)*

Prueba: Ver Billand y Bravard (2004: 606).

Ahora continuamos en el capítulo con la introducción a los algoritmos genéticos y su relación con la teoría económica, más precisamente, con el estudio de la evolución de juegos de formación de red.

Capítulo 3. Agentes y algoritmos genéticos

3.1 La modelación basada en agentes

El modelización basada en agentes es una forma computacional de modelar la interacción entre agentes. Sus resultados no son formales sino numéricos. Los agentes resultan ser bajo este marco de investigación códigos u objetos de programas. Los objetos agente tienen estados y reglas de comportamiento. Correr un modelo de agentes comprende la inicialización de la población de agentes, el proceso subsiguiente de interacción entre los mismos y el monitoreo del resultado final. En definitiva, correr el modelo en el tiempo la suficiente cantidad de veces, en caso de ser estocástico, es lo único que se necesita para 'resolverlo'.

Las ventajas y desventajas de la modelación basada en agentes respecto a los modelos formales son descritas por Axtell (2000). Como ventaja se especifica en primer lugar la flexibilidad de estos modelos para abarcar, por ejemplo, el comportamiento de agentes racionales o, mediante simples transformaciones en el código, el comportamiento de los mismos agentes pero bajo racionalidad acotada. Del mismo modo resulta relativamente

sencillo analizar una comunidad de agentes homogéneos ó, mediante una modificación en el proceso de inicialización, de agentes heterogéneos, con lo cual no es necesario recaer en la a veces excesiva simplificación formal de suponer un agente representativo para toda la población del análisis.

Asimismo, en el estudio de los procesos sociales importan tanto el espacio como las redes sociales de los sujetos. Matemáticamente resulta aún complejo tratar ambos ítems excepto en formas altamente estilizadas. Ambos problemas, sin embargo, son fácilmente tratables bajo la implementación de un modelo de agentes.

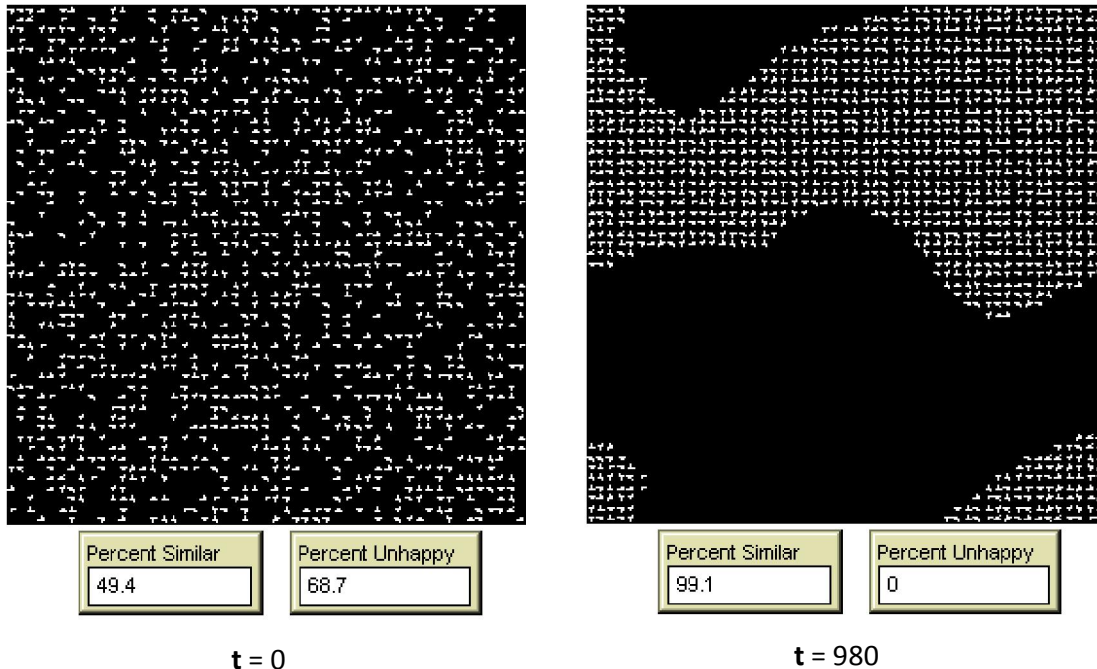
La gran desventaja de los modelos basados en agentes con respecto a la modelación matemática puede verse con respecto a la robustez de los resultados observados. Matemáticamente existen numerosas técnicas para derivar la robustez de un modelo formal explicitado pero en términos computacionales este proceso resulta menos claro. Usualmente en los procesos de simulación se ataca esta dificultad realizando múltiples corridas variando en cada serie de corridas los valores de parámetros sensibles al resultado final del modelo y luego observando cómo éstos diferentes resultados evolucionan en el tiempo.

El enfoque basado en agentes, de este modo, se presenta como una alternativa de análisis para diversos casos de estudio de las ciencias sociales en general y de la Economía en particular. Un aspecto básico de este enfoque es que construye sus modelos de lo individual hacia lo agregado (*bottom-up*) lo cual resulta ser al revés de los modelos formales clásicos aplicados en Economía los cuales parten de un agregado (agente representativo) para llegar

a deducir cuestiones a escala individual (*top-down*). De hecho, un resultado general en estos modelos que ha dado gran impulso a su utilización es lo que se denomina surgimiento (*emergence*) de conductas agregadas complejas partiendo de conductas individuales simples (Schelling, 1978, puede identificarse como el inicio de toda esta literatura). En ese sentido, el modelo de Schelling de segregación plantea a cada individuo como un agente que simplemente se localizará en un espacio virtual si cierta proporción, que puede hacerse variar, de sus vecinos es igual a su 'tipo' (la simulación inicial parte de modelar solo 2 'tipos' de agentes), mientras ello no ocurra el agente se 'mudará' constantemente de su ubicación actual hasta satisfacer su función de utilidad. Al final de un número determinado de iteraciones del modelo el espacio virtual resultante deja observar aglomeraciones de agentes de un mismo tipo, al estilo de los observados en los ghettos ó barrios culturalmente homogéneos en un plano urbano real. Este simple ejercicio de simulación resultó ser un aporte muy importante para muchos sociólogos interesados en temas de movilidad urbana y conformación de vecindarios así como en temas más generales de segregación. La Figura 3 presenta el desarrollo de un ejemplo del modelo de Schelling realizado en un paquete dedicado de simulación de agentes.

Dado que mayormente el sujeto de estudio es el resultado de programas basados en objetos, la ciencia de la computación en especial las áreas aplicadas a simulación y programación orientada a objetos resulta trascendental en secciones muy sensibles del análisis. En forma resumida, para construir un modelo económico basado en agentes se pueden aplicar las técnicas de resolución de problemas:

Figura 3. Modelo de segregación de Schelling (1978)



Simulación realizada en NetLogo 1.2.1, comprendiendo 2500 agentes (cada agente es un triángulo y hay dos clases de agentes, negros y blancos sobre un fondo negro). El porcentaje de similitud representa la cantidad deseada por cada agente de vecinos similares a sí mismo. El porcentaje de infelicidad representa la cantidad de agentes cuyos vecinos son predominantemente, en relación con el porcentaje de similitud, diferentes a él. Luego de 980 iteraciones se aprecia muy claramente la conformación de vecindarios homogéneos entre sí. La simulación se detuvo cuando el porcentaje de infelicidad llegó a cero.

- 1) se debe delimitar bien el problema de estudio y el marco sobre el cual éste es relevante, para ello se recurre, en este caso, a la teoría económica, y
- 2) se debe implementar la solución de manera que refleje lo mas acabadamente el marco teórico sobre el cual se sustenta haciendo hincapié en las posibles implementaciones alternativas y el grado de eficiencia respectiva, para lo cual se recurre las ciencias de computación.

En general se inicia con el diseño de pseudo-códigos y variables relevantes a ser observadas, realizando trazas y ensayos que permitan validarlos y luego se concluye con el diseño de un

programa, el cual depende de la implementación del lenguaje que se elija, debiéndose probarse a fin de que realice todo lo que al autor precise. En definitiva este programa será el ‘experimento’ que brindará respuestas a las preguntas para los que fue diseñado a tratar.

Tabla 1. Pseudo-código para el Juego de Red de Comercio (Trade Network Game)

```

int main () {
    Init(); // Construir generación inicial de comerciantes
           // con estrategias aleatorias.
    For (G = 1,...,GMax){ // Ingresar el lazo del ciclo de generación.
        // Ciclo de Generación:
        InitGen(); // Configurar comerciantes con parámetros
                 // proporcionados por el usuario (niveles de utilidad
                 // esperada, cuotas de capacidad,...).
        For (I = 1,...,IMax) { // Ingresar el lazo del ciclo de intercambio.
            // Ciclo de Intercambio:
            MatchTraders(); // Determinar los socios comerciales,
                           // dada las utilidades esperadas,
                           // y guardar las búsquedas de trabajos así como
                           // los costos de inactividad.
            Trade(); // Implementar los intercambios y guardar
                   // los pagos comerciales.
            UpdateExp(); // Actualizar las utilidades esperadas con
                       // pagos y costos recientes.
                       // Paso Ambiental:
            AssessFitness(); // Evaluar los resultados de ajuste del comerciante.
            Output(); // Información de comerciante sale por resultado.
                     // Paso de Evolución:
            EvolveGen(); // Evoluciona una nueva generación de comerciantes.
                       }
                       Return 0;
                       }
}

```

Un esquema de modelación basada en agentes que negocian en términos de sus propias funciones de utilidad es presentado por Testfatsion (1999). En las Tabla 1 y Tabla 2 se aprecia el pseudo-código del juego de red de intercambio planteado por la autora. Allí se observa la modularidad del diseño, lo que favorece la realización de experimentos con diferentes especificaciones de estructuras de mercado, búsqueda y encuentro (*matching*) entre comerciantes, formación de expectativas y actualización de la información así como evolución de las estrategias comerciales adoptadas por los agentes.

Tabla 2. Un mercader del juego TNG como agente de software

```
class TradeBot
{
  Normas Sociales Internalizadas:
  Protocolos de mercado para comunicarse con otros mercaderes;
  Protocolos de mercado para buscar y encontrar otros mercaderes;
  Protocolo de mercado para interacciones.
  Información de estado almacenada internamente:
  Mis atributos;
  Mi dotación inicial;
  Mis creencias y preferencias;
  Direcciones mías y de otros comerciantes;
  Información adicional que poseo acerca de otros mercaderes.
  Reglas de conducta internas:
  Mis reglas para buscar y procesar nueva información;
  Mis reglas para determinar mis reglas de conducta;
  Mis reglas para actualizar mis expectativas y preferencias;
  Mis reglas para medir mi nivel de preferencia (ajuste);
  Mis reglas para modificar mis reglas.
};
```

Asimismo, cada agente puede ser representado por una clase con normas sociales internalizadas, información de estado internamente almacenada y reglas de conducta internas. Los mercaderes, para este caso, pueden realizar conductas anticipadas usando las direcciones almacenadas junto con las normas de comunicación internalizadas. Ello daría pie a que los agentes puedan comunicarse entre ellos en situaciones provocadas por eventos específicos, una característica no presente en los modelos económicos estándares. Esta descripción recalca, finalmente, una de las características básicas de la definición de agente: su autonomía.

La modelación debe tener en cuenta, asimismo, si el ambiente sobre el cual los agentes desarrollaran sus acciones cumplimenta lo establecido por la teoría. En caso de comparar diferentes ambientes (*landscapes*), ello debe tenerse en cuenta cuando se analizan los resultados finales. Aquí la comparación entre el efecto que produce cada ambiente en

particular puede ser algo difícil si no se tiene una idea preestablecida del posible signo y dirección de los cambios.

En términos de la creciente importancia de la simulación basada en agentes, pueden tenerse en cuenta las siguientes referencias. Un buen trabajo introductorio sobre como la implementación basada en agentes puede ir depurando el entendimiento de situaciones de análisis cada vez más complejas puede verse en Gulyás (2001). Como trabajos aplicados en otras áreas de la ciencia pueden presentarse a Berger (2001) en agricultura, Cederman (2001) en estudios internacionales (en el caso específico de surgimiento de conflictos armados), Cohen y otros (1998, 2001) para el estudio de conductas que incentiven la cooperación de los agentes, mientras que Johnson (2002) relata su experiencia con modelos basados en agentes que ayudan a comprender la complejidad de los mercados financieros. Una aplicación a la dirección y planificación de proyectos de infraestructura es presentada en North (2001), Brent y Thompson (1999) y Brent et al. (2000) para la descripción de los potenciales usos de la modelización basada en agentes para comprensión, comparación y enseñanza de fenómenos sociológicos, entre otros. Un excelente trabajo sobre los cambios que se vienen observando en el estudio de las ciencias sociales debido a la importancia de la simulación basada en agentes puede leerse en Macy y Willer (2002), quienes confirman la importancia de la subdisciplina y de cómo estos modelos han desplazado en algunos campos a las técnicas estadísticas clásicas (los 'factores' del título del trabajo) en la explicación de los fenómenos sociológicos.

Otra forma de representar a los agentes o sus estrategias es a través de algoritmos genéticos. La siguiente sección profundiza en esta definición.

3.2 Algoritmos genéticos como agentes en la economía

El algoritmo genético (AG) está inspirado en los mecanismos basados en la evolución biológica y su transferencia a la búsqueda de soluciones para el problema que se pretenda resolver. Producto de la evolución biológica mediante la selección de genotipos (combinados con los cromosomas), basada en la adaptación a su medio ambiente que generan los fenotipos (la calidad de esta adaptación se mide por el rendimiento - fitness - en cada genotipo). Los genotipos que mejor se adapten a su entorno tienen una mayor facilidad de reproducción y la reproducción sexual garantiza el cruce de los de mejor rendimiento en los genes de la población. Los errores de las copias de los cromosomas durante la reproducción también introducen la novedad en la forma de mutaciones en los genes y sus consecuencias para la equiparación fenotipo. La selección continúa en la población de los genotipos con el rendimiento en el mayor llevado a una mejor adaptación de la población con su entorno (esto corresponde a un aumento en el rendimiento promedio de la población). El Apéndice 1 presenta un pequeño glosario de términos genéticos.

Los AG pretenden replicar este proceso. La relación entre los genotipos y fenotipos son entonces de codificación que tiene para el espacio de los parámetros del problema que pretenden resolver. Los valores posibles de una variable real o entero se codificarán por ejemplo, como una cadena binaria, equivalente a los cromosomas de los genotipos), que consiste en 0 y 1. En su versión más sencilla, una variable, el algoritmo genético va a evolucionar una población de cromosomas inicializan aleatoriamente, basado en el rendimiento relativo de cada uno en el problema considerado. El cruce y la mutación se produce al mismo tiempo en esta población de cromosomas para permitir la exploración de

los valores espaciales de las variables consideradas (véase la sección siguiente para un ejemplo y una presentación más detallada).

Contribuciones de algoritmos genéticos a estudios de economía hay varios en la literatura (Nisan et al. (2007), Wang et al. (2002); Cohen et al. (1998, 2001); Alkemade (2004); Özyildirim (1997); Chen y Ni (2000), Alemdar y Özyildirim (1998); Angus (2006); Axelrod (1997); Axtell (1999, 2000, 2003), Pruijt (2000), Marks (1998, 2000)), Iba (1998). Pocos se han focalizado en aplicarlo en juegos de formación de red, siendo una excepción los trabajos de Carayol y Roux (2005) y Carayol et al. (2005, 2008). Estos autores estudian principalmente juegos con grafos no dirigidos (el modelo de coautor, entre otros) compendiados en Jackson (2005) y Goyal (2008). Los resultados de sus contribuciones destacan la bondad de los AG para procesar combinatorialmente una enorme potencialidad de soluciones alternativas. Asimismo, plantean los problemas de computabilidad inherentes al estudio de redes óptimas (Airoidi (2005), Nisan et al. (2007), Tardos y Wexler (2007)). Otro trabajo en este estilo que puede ser citado es Cajueiro (2005) quien testea primitivos modelos de formación de red, no utilizados en ese trabajo.

3.3 Los algoritmos genéticos

Los algoritmos genéticos serán la herramienta de evaluación de esta investigación, por lo que a partir de esta sección estará dedicada a introducirlos³. Puntualizaremos su utilización como una metáfora del aprendizaje social. En ese sentido es un aporte a la línea iniciada por Riechmann (1999). Como ya ha sido demostrado por literatura el aprendizaje por AG se compone de tres diferentes esquemas aprendizaje. La combinación de estos tres esquemas

³ Esta sección está basada en Grefenstette y Baker (1994), Vallée y Yildizoglu (2003) y Riechmann (1999, 2001).

es lo que le da al aprendizaje por AG una clase de estabilidad entre medio de la convergencia asintótica y la explosividad.

En este proceso, también se definirán los principales componentes y subcomponentes de un algoritmo genético (AG) clásico y las variaciones que se presentan para el caso de estudio en particular. Particular interés se presenta en la definición de la función de ajuste y en la función de generación de poblaciones.

Se verá que:

- Los AG optimizan el *trade-off* entre explorar puntos nuevos en el espacio de búsqueda y la explotación de la información descubierta hasta ese momento.
- Los AG tienen la capacidad de paralelismo implícito, lo que quiere decir que el efecto de AG es equivalente a una búsqueda extensiva de hiperplanos dados un espacio dado, sin directamente comprobar todos los valores de los hiperplanos. Cada esquema denotaría un hiperplano.
- Los AG son algoritmos aleatorizados, en los cuales usan operadores cuyos resultados son gobernados por probabilidad.
- Los AG operan en muchos resultados simultáneamente, buscando información de los actuales puntos de búsqueda hacia la subsiguiente sección de búsqueda. Su capacidad para mantener múltiples soluciones concurrentemente hace de los AG menos susceptibles a los problemas de ruidos y máximos locales.

El aprendizaje por algoritmo genético es un modo aprendizaje *social* más que *individual*. Del aprendizaje social siempre implica aprender de los otros por ello que no existe un aprendizaje por algoritmo genético de un solo individuo aislado o de agentes aislados. Como se ha recalado en otras contribuciones existe una conexión cercana entre el aprendizaje por

algoritmo genético y la teoría económica evolucionista, dada que en ambos recaen fuertemente en el concepto de población en vez del agente económico como un ente aislado.

El aprendizaje social es estudiado como la búsqueda de información de acuerdo a reglas simples lo cual en el caso de los algoritmos genéticos implica el uso de tres diferentes técnicas: el aprendizaje por imitación (selección/reproducción), aprendizaje por comunicación (cruzamiento) y el aprendizaje por experimentación (mutación). Uno de los desafíos científicos para el aprendizaje por medio de AG es el encontrar si existen ciertas propiedades de los AG que llevan a los modelos de aprendizaje genético por lo menos a las cercanías de los resultados de los modelos económicos más representativos.

3.3.1 Esquema y definiciones

Un algoritmo genético implementa una versión muy simplificada y muy esquemática de los mecanismos de la evolución biológica. A partir de un problema

1. *Individuo/cromosoma/secuencia*: una solución potencial del problema que corresponde a un valor codificado de la variable (o de las variables) en consideración. En nuestro caso, la codificación deberá representar la estructura de conexión de una red.
2. *Población*: una ensamble de cromosomas o de puntos del espacio de búsqueda. En el caso del presente problema, será un conjunto inicial aleatorio de cromosomas sobre el cual trabajarán los operadores genéticos.
3. *Ambiente*: el espacio de búsqueda (caracterizados en términos de performance correspondiente a cada individuo posible). Será el espacio de conexiones codificadamente representadas en todas sus combinaciones posibles.

4. *Función de performance*: la función – positiva – que se busca maximizar dado que el representa la adaptación del individuo a su ambiente. Enmarcado en la literatura económica, la función de performance representará los beneficios de cada estructura de conexiones.

A un nivel muy general, el funcionamiento de un AG está también basado en las fases siguientes (ver Tabla 3)

1. **Inicialización**. Una población inicial de N cromosomas es obtenida aleatoriamente.
2. **Evaluación**: Cada cromosoma es decodificado y luego evaluado
3. **Selección**. Creación de una nueva población de N cromosomas por la utilización de una método de selección apropiado.
4. **Reproducción**. Posibilidad de cruzamiento y de mutación en el seno de la nueva población
5. **Retorno** a la fase de evaluación (hasta tanto las condiciones de finalización del problema no sean satisfechas).

Los algoritmos genéticos resultan de una emulación del proceso de evolución natural de las especies sobre algoritmos de selección y cruzamiento de una población inicial cuyo objetivo final es satisfacer un criterio de investigación. El trabajo seminal de Holland (1975) ha también evolucionado hacia esquemas de tratamiento de la información más amplios y flexibles.

Tabla 3. Pseudo-algoritmo genético**Procedimiento AG**

```

Empezar
   $t = 0$ 
  inicializar  $P(t)$ ;
  evaluar estructuras en  $P(t)$ ;
  mientras no verifique condición de terminación
  hacer
    Empezar
       $t = t + 1$ 
      seleccionar  $P(t)$  de  $P(t-1)$ ;
      cruzar estructuras en  $P(t)$ ;
      mutar estructuras en  $P(t)$ ;
      evaluar estructuras en  $P(t)$ ;
    Fin
  Fin.

```

En términos del lenguaje de programación de Netlogo, la correspondencia entre la simbología del capítulo 2 y el código se observa en Tabla 4.

Tabla 4. Correspondencia entre símbolos del modelo formal al computacional

Símbolo en el modelo	Símbolo en el código
n	num-nodes
v_i	own-information
$N^{i:g}$	genome
$\Pi_i(g)$	indirect-info
$\sum_{j \neq i j \in g} I_j(N^{i:g})$	profits
$I_i(g)$	indirect-information
c	direct-information + indirect-information
$\sum_{j \in N^{i:g}} \mu_j(g)$	edge-cost
δ	direct-edging-costs + indirect-edging-costs
$d(i,j;g)$	rate-of-decay
γ_0	indirect-info
γ	gamma_0
α	gamma
φ	alpha
q_i	phi
	production

El Apéndice 3 realiza una pequeña introducción al lenguaje de programación del entorno Netlogo. La inicialización y la ejecución del código están ejemplificados con los Código Netlogo 1 y Código Netlogo 2.

Código Netlogo 1. Procedimiento de inicialización

```
Entrada: generations (número de generaciones de la corrida, variable global ingresada por slider);  
Salida: una población aleatoria de cromosomas siendo presentado (build-ga-network) sólo el de mejor performance.  
  
;; Defino las razas de agentes que actuarán como nodos.  
breeds [node nodes]  
  
;; Defino las variables globales  
globals [genome-list]  
  
;; Defino las variables que pertenecen a los nodos  
nodes-own [genome]  
  
to setup  
  ca ;; Limpiar todo el ambiente  
  repeat generations * .02 [ ;; el 2% del total de generaciones se utiliza como  
    create-initial-genetic-info ;; población inicial de donde iniciar el proceso.  
    sort-global-fitness ;; ordeno las topologías y fitness por mayor fitness  
    build-ga-network item 0 topology-list ;; se representa la topología de mayor fitness  
    set best-fitness item (num-nodes * num-nodes) item 0 topology-list  
  ]  
end
```

El procedimiento de ejecución del algoritmo presenta la condición inicial de detención del algoritmo genético (sólo se detendrá cuando alcance el máximo de generaciones establecidos). Mientras ello no ocurra se llama secuencialmente a los procedimientos de selección y cruzamiento (*mate-recombine*), mutación (*mutate*) y repoblación (*rebirth*).

Código Netlogo 2. Procedimiento de ejecución

```
Entrada: generations (número de generaciones de la corrida);  
Salida: evolución de una población inicial aleatoria de cromosomas a través de una cantidad generations de generaciones.  
  
to go  
  set generation (generation + 1)  
  if (generation > generations) [stop] ;; cuando se alcance el máximo de  
  do-best-of-age-plot ;; generaciones, se detiene.  
  mate-recombine  
  mutate  
  rebirth  
end
```

Una población incluye varios cromosomas y el AG construye una nueva población a través de una reproducción selectiva (etapa **b**), los cruzamientos (combinación de soluciones existentes – etapa **d**) y las mutaciones (experiencias aleatorias, introducción de novedades – etapa **e**).

Código Netlogo 3. Procedimiento de selección

```
Entrada: topology-list (lista de la topología y performance de la población);  
Salida: seed-genomes (genoma de la población seleccionada).  
  
to mate-recombine  
;; Aquí tengo que elegir entre cual proceso de selección adopto para generar la nueva  
;; población.  
  if Selection = "Elitism" [  
    elitism  
    set seed-genomes item 0 winner-list]  
  if Selection = "Roulette" [  
    wheel-roulette  
    let chosen-generation item random-float length ranking ranking  
    let chosen-topology position chosen-generation ranking  
    set seed-genomes item chosen-topology topology-list ]  
  if Selection = "Tournament" [  
    tournament  
    set seed-genomes item 0 winner-list]  
;; Aquí se permite el cruzamiento de los elegidos por el proceso de selección previo  
make-mix-genes          ;; proceso de recombinación de genes  
let o 0  
while [o < (num-nodes * num-nodes)]  
[ set seed-genomes (replace-item o mix-genes item o seed-genomes)  
  set o o + 1 ]  
end
```

3.3.2 Codificación y población inicial

Primeramente, se deben representar los diferentes estados posibles de la variable donde se va a buscar al valor óptimo bajo una forma utilizable por un AG: esto se denomina *codificación*. Esta permite establecer una conexión entre los valores de las variables y los individuos de la población, de manera de imitar la conexión que existe en biología entre el genotipo y el fenotipo. Existe dos tipos de codificación principalmente: la codificación binaria (codificación bajo la forma de una cadena binaria) y codificación real (representación directa

de los valores reales de las variables). Se puede pasar fácilmente de una hacia la otra (Michalewicz, 1992). La codificación inicialmente utilizada por Holland (1985)

Definición 1. *Secuencia/Cromosoma/Individuo (Código binario)*

Denominamos una secuencia (cadena, cromosoma, individuo) A de longitud $l(A)$ una secuencia $A = \{a_1, a_2, \dots, a_l\}$, $\forall i \in \{1, \dots, l\}$, $a_i \in V = \{0, 1\}$.

Un cromosoma es un conjunto de bits (formado de unos y ceros) llamada así cadena binaria. Si la variable de estudio posee cuatro estados posibles se la puede codificar con una cadena binaria de longitud $l = 2: \{00, 01, 10, 11\}$. Por consiguiente, una cadena binaria de longitud l puede representar 2^l estados de la variable. El ejemplo del Ejemplo 1 supone que hay $2^{16} = 65536$ estados de la variable x .

En el caso particular de una codificación binaria, una función de *decodificación* d debe permitir el pasaje de la representación binaria a través de la representación en términos de los estados de la variable inicial. Si esta variable toma los valores enteros, debemos tener $d: \{0, 1\}^l \rightarrow N$ (ó l es la longitud de la cadena).

3.3.2.1 Representando redes como cadenas binarias

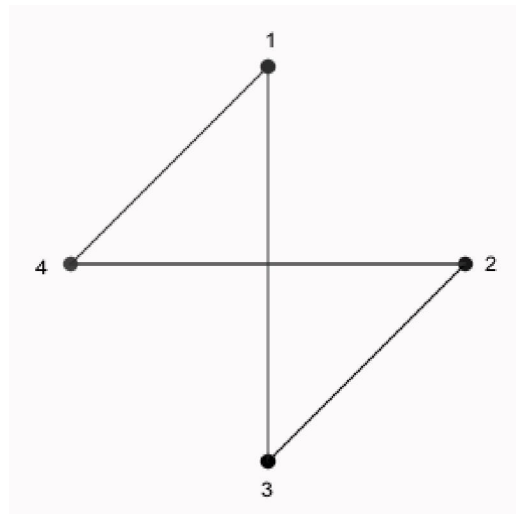
Nuestro problema es encontrar una red g la cual maximice el valor social Π dado por la ecuación 2.3.3 sobre el conjunto de potenciales redes G . A fin de usar los AG para este problema de optimización, necesitamos representar las redes como cadenas de caracteres binarios (secuencias de bits – 1 ó 0).

Considérese primero que cualquier red con n agentes (sea dirigida o no) puede, sin pérdida de generalidad, ser representada por una matriz de conexiones de tamaño $n \times n$ de elementos binarios. En todas las redes que consideramos las auto-conexiones están excluidas, por lo que la matriz de adyacencia, excluyendo la diagonal, provee la información completa de la estructura de red.

Como una consecuencia, el vector compuesto por todos los bits de conexiones de esta parte supra triangular en algún orden convencionalmente elegido determina la estructura de la red. Por ello, para una red de n agentes, este vector es la cadena binaria de longitud l .

Ejemplo 1. Una aplicación de los AG para la evolución de una red. La red originalmente es representada por la una estructura aleatoria inicial de conexiones correspondiente a todos los agentes (nodos) en la red. Supongamos una red de 4 agentes o jugadores. Por ejemplo, cada jugador puede elegir tirar una moneda y dependiendo de si sale cara o ceca conectarse o no con cualquiera de los 3 agentes restantes. Del mismo modo cada agente, a su vez, decide conectarse aleatoriamente con sus otros colegas. Un resultado puede ser el siguiente.

Figura 4. Ejemplo 1



Esta red puede ser representada por una matriz de adyacencia. En ella, cada fila representa las decisiones de conexión de cada jugador y cada columna representa la decisión de conexión correspondiente con cada jugador. Ello se puede apreciar en A_1 .

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Esta misma información se puede representar en un vector booleano del tipo $A_1 = [0011000101000100]$ de longitud $n \times n$. Vamos a considerar que cada matriz de adyacencia representa un cromosoma de la población.

En el caso de una red de comunicación bidireccional, en la cual el establecimiento de un enlace debe hacerse de mutuo acuerdo entre el origen y el destino, la representación natural es un grafo no dirigido. La matriz de adyacencia, para este caso particular, es simétrica. Sólo se les debe pedir a los agentes crear la diagonal superior de la matriz de adyacencia. Por ejemplo, para el caso del Ejemplo 1, de ser modelada como una relación de comunicación bidireccional, la matriz de adyacencia quedaría representada como:

$$\bar{A}_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Obsérvese que la diagonal inferior de la matriz \bar{A}_1 es simétrica a la diagonal superior. Por ello, para este caso particular, sólo es necesario crear un vector booleano que represente a la diagonal superior. Así, el vector booleano $\bar{A}_1 = [011110]$ representa la información de la matriz de adyacencia superior y es de longitud $(n-1)!$

El Código Netlogo 4 muestra como se crea la población inicial en el código ensayado por este trabajo.

Código Netlogo 4. Creación de la población inicial

```

Entrada: nodes (num-nodes nodos creados previamente);
Salida: genome-list (matriz de adyacencia como lista de vectores).

to create-genome
;; Se le pide a los nodos que inicialicen un listado genome como un listado vacío para luego
;; llenarlos con los valores aleatorios binarios en cualquier orden y en número igual al
;; de nodos (num-nodes). El ítem de la lista correspondiente al propio nodo es fijado en cero,
;; para evitar autoconexiones.

ask nodes [
  set genome [] ;; creo una lista vacía y luego la lleno con num-nodes valores
  set genome n-values num-nodes [random 2] ;; aleatorio booleanos
  set genome replace-item who genome 0 ;; reservando el propio valor con un cero
]

;; se crean la lista de genomas (genome-list) desde la cual se va a
;; crear la matriz de adyacencia
let i 0
while [i < num-nodes]
[set genome-list lput [genome] of node i genome-list
set i i + 1]

end

```

3.3.3 Función de performance

Cada cromosoma aporta una solución potencial al problema a resolver. De todos modos, esta solución no tiene el mismo grado de relevancia. Le corresponde a la función de performance (*fitness*) medir la eficacia para permitir al AG de hacer evolucionar a la población en beneficio de encontrar la mejor solución. Dicho de otro modo, la función de performance $f(\cdot)$ debe poder atribuir a cada individuo un indicador positivo representando su pertinencia para el problema que se busca resolver.

En nuestro caso de estudio, el valor de la función objetivo ($\Pi(g) = I(g) - c(g)$) puede ser utilizado directamente porque los valores necesarios para su estimación se encuentran. De manera más general.

Definición 2. *Performance – fitness – de una secuencia.* Representamos la performance de una secuencia A por una función positiva, $f : A \rightarrow f(A) \in R_+$.

En el caso de una codificación binaria, la función de performance debe afectar un valor positivo en la codificación correspondiente (fenotipo) a cada cadena binaria (genotipo): $\forall A \in \{0,1\}^l, f(d(A)) \in R_+$. La función de un algoritmo genético es entonces simplemente encontrar la cadena que maximiza esta función f . Evidentemente en el caso de una codificación binaria, cada problema particular necesitará sus propia función d y f .

Una vez que la performance de cada individuo en la población actual es evaluada, los mecanismos evolutivos entran en juego para proceder a la exploración del espacio de valores. Los operadores del AG intentar imitar estos mecanismos.

3.3.3.1 Beneficios directos

Los beneficios directos se obtienen de evaluar la información directa obtenida y el costo directo de la conexión del nodo i al nodo j . En ese sentido, la fuente básica de información de las conexiones directas es la propia matriz de adyacencia. Si el nodo i (fila i de la matriz de adyacencia) presenta en la columna j un 1 en su matriz de adyacencia entonces el nodo i accede a la información que posee el nodo j y paga un costo de una unidad de conexión.

3.3.3.2 Beneficios indirectos

Para calcular la acumulación de costos indirectos debe conocerse que recorrido realizó la información para poder tasar el costo de transporte de la misma desde el origen, pasando por todos los destinos intermedios hasta llegar al destino final. Para ello se requiere, en términos de teoría de grafos, calcular la clausura transitiva del grado dirigido (no dirigido). Esta clausura transitiva indica todos los caminos de menor recorrido que unen a un origen con su destino. Una de las formas de estimar la clausura transitiva de una matriz de adyacencia es utilizando el algoritmo de Floyd-Warshall, procedimiento que fue adoptado en este trabajo (ver Apéndice). Para el Ejemplo 1 la matriz de clausura transitiva A_1^T se aprecia a continuación.

$$A_1^T = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 3 & 4 \\ 0 & 1 & 3 & 4 \end{bmatrix}$$

Por ejemplo, el camino mínimo entre el nodo 3 y el nodo 4 es de 3. Es decir, la información debe pasar tres nodos antes de llegar finalmente desde el nodo 3 al nodo 4. El Código Netlogo 5, a continuación, presenta el extracto del código utilizado para calcular los beneficios directos e indirectos de los nodos de cada matriz utilizados para los modelos basados en Bala y Goyal (2000) y Larrosa y Tohmé (2003b). Las entradas para los códigos son

el número de nodos (`num-nodes`), el vector de conexiones individual (`genome`) derivado de la matriz de adyacencia, el vector de información indirecta (`indirect-info`) que se deriva de la matriz de alcanzabilidad (Floyd-Warshall) y la tasa de decaimiento de la información (`rate-of-decay`). El Apéndice 2 al final desarrolla el algoritmo de Floyd-Warshall implementado en este trabajo.

Código Netlogo 5. Procedimiento de cálculo de beneficios (fitness) para modelos de formación de red basado en Bala y Goyal (2000) y Larrosa y Tohmé (2003b, 2005)

```

Entrada: genome (lista de conexiones de cada nodo), indirect-info (lista de clausura transitiva de cada nodo), rate-of-decay (tasa de decaimiento de la información), own-information (valor de la información poseía por cada agente, estandarizada a 1) y ratio-info-edge (determina el valor relativo de la información respecto al costo de conexión);
Salida: profits (definido como el valor de toda la información de toda la sociedad menos los costos sociales de conexión)

to calculate-profits
;; Procedimiento delegado a los nodos dado que se utilizan variables de su dominio
ask nodes [
  set direct-edging-costs 0
  set direct-information own-information
  set indirect-information 0
  set indirect-edging-costs 0
  set net-direct-benefit 0
  set net-indirect-benefit 0
  set profits 0
  set edge-cost own-information / ratio-info-edge
]

;; itero sobre los nodos y les inquiero sobre sus costos y beneficios de información "directos"

let j 0
let k 0
while [j < num-nodes]
[set k 0
  while [k < num-nodes]
  [ask node j [if (item k genome = 1)
    [ifelse decay [set direct-information direct-information + (rate-of-decay *
      [own- information] of node k)]
    [set direct-information direct-information + [own-information] of node k]
    set direct-edging-costs direct-edging-costs + edge-cost ]
    set k k + 1]
  ]
  set j j + 1]

;; se itera sobre los nodos y se les pregunta por sus costos e información "indirectos"

set j 0
set k 0
while [j < num-nodes]

```

```

[set k 0
while [k < num-nodes]
  [ask node j [ifelse (item k indirect-info > 0)
    [ifelse decay
      [set indirect-information indirect-information + ([direct-information] of node k *
        (rate-of-decay ^ item k indirect-info))]
      [ set indirect-information indirect-information + [direct-information] of node k]
      set indirect-edging-costs indirect-edging-costs + (item k indirect-info * edge-cost)]
    [ set indirect-information indirect-information
      set indirect-edging-costs indirect-edging-costs]
    set k k + 1]]
  set j j + 1]
ask nodes [
  if edge-Cost-Structure = "direct cost direct info"
    [set indirect-information 0 set indirect-edging-costs 0]
  if edge-Cost-Structure = "direct cost indirect info"
    [set indirect-edging-costs 0]
  if edge-Cost-Structure = "indirect cost direct info"
    [set indirect-information 0]
  if edge-Cost-Structure = "indirect cost indirect info"
    []
]

;; Ahora sumamos todos los ingresos y costos individuales para obtener el beneficio social

set net-direct-benefit sum [direct-information] of nodes - sum [direct-edging-costs] of nodes
set net-indirect-benefit sum [indirect-information] of nodes -
  sum [indirect-edging-costs] of nodes
set profits precision (net-indirect-benefit + net-direct-benefit) 3
end

```

El Código Netlogo 6 hace lo propio para los modelos de Goyal y Joshi (2003) y Billand y Bravard (2005).

Código Netlogo 6. Procedimiento de cálculo de beneficios (fitness) para modelos de formación de red basado en Goyal y Joshi (2003) y Billand y Bravard (2005)

```

Entrada: alpha, gamma, gamma_0, phi, num-nodes (variables definidas en la definición formal de cada modelo);
Salida: profits (definido como el valor de la suma todos la beneficios individuales de todas las firmas)

to calculate-profits
;; Procedimiento delegado a los nodos dado que se utilizan variables de su dominio
set production 0
set profits 0
set q2 0

;; Sumo todos los nodos realizados por todos las firmas para saber después el ni
let i 0
set sum-genome 0

```

```

while [i < num-nodes][
  set sum-genome sum-genome + sum [genome] of node i
  set i i + 1
]

;; Cumpló con la condición de producción positiva
let h 0
set condition-positive-production ((alpha - gamma_0) - gamma * (num-nodes - 1) ^ 2)

;; recorro todos los nodos para determinar cuánto producen de acuerdo a la función de
;; producción del paper.

while [h < num-nodes] [
  ifelse condition-positive-production < 0
  [set production 0]
  [set production production + ((alpha - gamma_0) + ((gamma * num-nodes * sum [genome] of
  node h) - (gamma * (sum-genome - sum [genome] of node h))) / (num-nodes + 1))]
  set q2 q2 + ((alpha - gamma_0) + ((gamma * num-nodes * sum [genome] of node h) - (gamma
  * (sum-genome - sum [genome] of node h))) / (num-nodes + 1)) ^ 2

  set profits profits + precision (production ^ 2 - (phi * sum [genome] of node h)) 1
  set h h + 1
]

;; Calculo los valores críticos para las topologías óptimas
set g_c (gamma * num-nodes) * ((2 * (alpha - gamma_0) - (gamma * num-nodes * (num-nodes -
3))) / ((num-nodes + 1) ^ 2)) ;; valores criticos para red completa
set g_e (gamma * num-nodes) * ((2 * (alpha - gamma_0) + (gamma * num-nodes * (num-nodes
+ 1))) / ((num-nodes + 1) ^ 2)) ;; valores criticos para red vacía
set gamma_critical (alpha - gamma_0) / (num-nodes - 1) ^ 2

end

```

Otro indicador de performance del algoritmo será el indicador de densidad de red. El mismo se define como la proporción de enlaces realizados sobre el total de enlaces potencialmente realizables.

Hay dos indicadores de densidad de red empleados en este trabajo. Para grafos dirigidos:

$$D_i = \frac{\sum_{\forall i \in G} N_i(g)}{n(n-1)}$$

Y para grafos no dirigidos es:

$$D_i = \frac{2\sum_{\forall i \in G} N_i(g)}{n(n-1)}$$

Airoldi et al. (2005) plantean, a su vez, otras métricas de redes.

3.3.4 Los operadores

Tres operadores juegan un rol preponderante en el posible funcionamiento de un AG: el operador de selección, el operador de cruzamiento y el operador de mutación. Si el principio de cada uno de estos tres operadores es fácilmente comprensible, es completamente difícil de explicar la importancia aislada de cada de los operadores en el funcionamiento del AG. Ello se debe al hecho que cada uno de ellos son marcados por propiedades que le son propias (valor selectivo de individuos, probabilidad de activación del operador, etc.)

3.3.4.1 Operador de selección

La optimización por búsqueda aleatoria es un procedimiento de optimización para evaluar un conjunto aleatorio de números y elegir el mejor. Los algoritmos genéticos son una clase de métodos de optimización por búsqueda aleatoria.

Supongamos que queremos resolver

$$\max_{x \in S} f(x), \quad (2.1)$$

donde $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ y S es una subconjunto de \mathbb{R}^n , de la forma $S = S_1 \times S_2 \times \dots \times S_n$, $S_i = [a_i, b_i]$. Dado que f es usada para modelar un ajuste, es natural que sea positiva. Si f es positiva en todo \mathbb{R}^n , entonces la reemplazamos con una función de la forma $\max\{0, f(x) - f(c)\}$ para algún $c \in S$ donde $f(c) < 0$. Tal reemplazo en (2.2) deja al problema básicamente sin cambios.

Un algoritmo genético observa a f en un conjunto de puntos, con ese conjunto de puntos evolucionando de tal modo de mantener en su interior valores altos de f . Se empieza por seleccionar m puntos de S para construir $X_1 = \{x^1, \dots, x^m\}$. Se construye luego una secuencia de dichos conjuntos; sea X_k el conjunto de x 's generados en la etapa k . Los puntos iniciales

podrían reflejar alguna información acerca donde los máximos de f son probables que se encuentren o puedan ser elegidos al azar. La elección de m es importante. Para m muy grandes harán la búsqueda más global y reducirán la probabilidad de una convergencia prematura, pero al costo de mayor cantidad de evaluaciones sobre f .

La idea básica es evaluar f para cada $x \in X_k$. Los puntos x son llamados “cromosomas” y el valor de f en x es una medida del ajuste de x . Entonces se emparedan pares de $x \in X_k$ para producir nuevas cromosomas, tal emparede comprende tanto la “recombinación” como la “mutación”. Algunos de estos nuevos cromosomas entonces remplazan a algunos de los cromosomas iniciales para crear una nueva población, X_{k+1} . El algoritmo produce una secuencia de poblaciones de este modo y finaliza bien cuando termina cuando se alcanza un número predeterminado de generaciones o cuando se tiene un x^* que parece ser tan bueno como es posible. Estos detalles se estipulan cuando se diseñan las reglas de emparede, el valor de la tasa de mutación y las reglas de reemplazo de los puntos viejos por los nuevos. A continuación se describirá la implementación utilizada en Dorsey y Mayer (1992).

Dado los valores de f en $x \in X_k$, queremos determinar el proceso de emparede. Queremos que aquellos x con valores altos de f sean “seleccionados”. Entonces tomamos algunos $T: \mathbb{R} \rightarrow \mathbb{R}$ positivos y crecientes y definimos

$$p_i = \frac{T(f(x^i))}{\sum_{x \in X_k} T(f(x))} \quad (2.3)$$

como la probabilidad que $x^i \in X_k$ sea elegido para participar. Dorsey y Mayer (1992) eligen $T(y) = y + a$, donde a puede ser alterado entre iteraciones para mantener la probabilidad p_i a fin de que no sea ni muy difusa ni muy concentrada. Usando las probabilidades en (2.2) se

realizan m extracciones de X_k con reemplazo, produciendo un conjunto Y_k de m cromosomas que participarán en el cruzamiento.

Ahora tomamos dos puntos y^1, y^2 de Y_k sin reemplazo. Queremos cruzar o empardar y^1 con y^2 para producir dos nuevos vectores, z^1 y z^2 . Para hacer eso, seleccionamos $i \in \{1, 2, \dots, n\}$ al azar y tenemos y^1 e y^2 "cruzados" en la posición i , esto es

$$\begin{aligned} z^1 &= (y_1^1, \dots, y_i^1, y_{i+1}^2, \dots, y_n^2), \\ z^2 &= (y_1^2, \dots, y_i^2, y_{i+1}^1, \dots, y_n^1) \end{aligned} \quad (2.4)$$

Seguimos extrayendo pares de y^1 e y^2 de Y_k sin reemplazo hasta haber mapeado todo Y_k .

Esta especificación del cruzamiento en (2.1) es sólo una forma posible de especificar el proceso de cruzamiento. Otra opción podría ser la de construir $z_i^1(z_i^2)$ sumando la parte entera de $y_i^1(y_i^2)$ a la parte fraccional de $y_i^2(y_i^1)$, $i = 1, \dots, n$. Hay una infinidad de formas de especificar el proceso de cruzamiento, con cada uno modificando con algún criterio el "esquema", es decir, la forma de codificar los puntos en S . Se puede ver a Holland (1975) para una teoría general y descripción de estos esquemas.

Hay numerosas alternativas para mejorar la implementación de un algoritmo genético. Examinamos la implementación de Dorsey y Mayer porque es simple. Los mismos autores aplican este algoritmo genético para replicar un variado grupo de estimaciones econométricas donde la estimación original utilizó procedimientos de optimización estándares. Sus replications usando AG en algunos casos encontraron resultados estadísticamente más significativos y con una interpretación económica diferente a los trabajos publicados.

Un enfoque más poderoso es el de métodos genéticos híbridos. En tales modelos, un empieza con una colección de puntos y los cruza para producir un nuevo conjunto pero entonces permite un procedimiento de optimización estándar para proceder con un número fijo de iteraciones para cada uno de estos puntos. Este produce una nueva colección de padres que serán cruzados y así subsiguientemente. La ventaja de los métodos híbridos es que el procedimiento estándar se moverá hacia los mejores óptimos locales. En este acople de métodos estándares y genéticos se observa que el componente genético esencialmente provee de un enfoque más sistemático para reiniciar las conjeturas aleatorias.

Este operador determina la capacidad de cada individuo a persistir en la población y a diluirse de ella. En regla general, la probabilidad de sobrevivir de un individuo será proporcionalmente relativa a su performance respecto de los otros individuos. Esta traduce bien la idea de la selección natural: los genes más beneficios tenderán a difundirse en la población mientras que aquellos con performance más relativa más pobre tenderán a desaparecer.

Existen muchos métodos para representar la reproducción. El método más común conocido y el más utilizado es sin dudas el de selección por ruleta (*roulette wheel*) de Goldberg (1989). Según este método, cada cromosoma será duplicado en una nueva población proporcionalmente a su performance. Se efectúa, en cualquier orden, tantas tiradas sin repetición como elementos haya en la población. De este modo, en el caso de una codificación binaria, la performance de un cromosoma particular, A_i , es $f(d(A_i))$, la probabilidad con la que éste será reintroducido en la nueva población de tamaño N es:

$$P[A_i] = \frac{f(d(A_i))}{\sum_{j=1}^N f(d(A_j))}$$

La performance de un individuo es elevada por información a los de los otros individuos de la misma población por lo que tiene la probabilidad de ser reproducido en la población. Los individuos que poseen una gran performance relativa tienen más chance de ser seleccionados. Hablamos ahora de la *selección proporcional*. El número esperado de cada individuo dentro de la nueva población que va a resultar de la selección es ahora igual a:

$$n_i = N \times P[A_i] = N \times \frac{f(d(A_i))}{\sum_{j=1}^N f(d(A_j))} = \frac{f(d(A_i))}{\frac{1}{N} \sum_{j=1}^N f(d(A_j))} = \frac{f(d(A_i))}{\bar{f}}$$

Luego que una tirada aleatoria es efectuada de acuerdo a sus números esperados de reproducción. Todo ocurre como si se arrojaran N flechas sobre un juego en forma de un queso donde la parte de cada individuo es igual a su performance relativa o como si fuese a girar una rueda de la fortuna de la misma forma. De allí es donde surge el nombre de este proceso de selección.

El inconveniente mayor de este método recae en el hecho que un individuo no es el mejor puede de todos modos dominar el proceso de selección. Este método puede también engendrar una pérdida de diversidad por la dominación de un individuo. Un tercer problema es que su débil performance cerca del fin del proceso cuando los individuos se parecen (porque la población está entonces dominada por los individuos tiene desempeños relativamente similares). Dawid (1999: 39) resume muy bien todos estos inconvenientes:

Piense en una situación donde una cadena (para nosotros un cromosoma) de la población posee comparativamente un desempeño elevado sin ser óptimo o muy próximo al óptimo.

Digamos que el desempeño de esta cadena es diez veces mayor que el desempeño medio [...] se podría fácilmente arribar, después de algunas generaciones, a que la población no esté más que enteramente constituida por esta cadena. En tal caso, el algoritmo genético no evolucionará más y el óptimo jamás será alcanzado. Este fenómeno se denomina “convergencia prematura” y este uno de los problemas más frecuentes en la utilización de los algoritmos genéticos.

El Código Netlogo 7 muestra el procedimiento de selección por ruleta utilizado. El procedimiento crea variables locales y luego crea una variable que acumulará los valores de fitness de todos los individuos [sum-fitness]. Esa será utilizada como referencia para crear la lista *ranking* la que presenta los individuos ordenados por la proporción de su fitness respecto al resto de los individuos.

Código Netlogo 7: Procedimiento de selección por ruleta

```
Entrada: seed-genome (lista de conexiones de cada nodo del pool de selección), topology-list (lista de matrices de adyacencia y su pago asociado);  
Salida: ranking (lista de los individuos elegidos por el procedimiento para apareamiento mediante la regla de ruleta)  
  
to wheel-roulette  
let i 0 ;; se crea una variable local i  
set sum-fitness 0 ;; se inicializa sum-fitness ...  
set ranking [] ;; y ranking  
while [i < min (list (generations * .01) length topology-list) ]  
  [ set sum-fitness sum-fitness + item (num-nodes * num-nodes) item i topology-list  
    set i i + 1 ]  
set i 0  
while [i < min (list (generations * .01) length topology-list) ]  
  [set ranking lput ((item (num-nodes * num-nodes) item i topology-list) / sum-fitness) ranking  
    set i i + 1 ]  
end
```

Brevemente, existen otros métodos de selección los más conocidos son los de torneo (*tournament selection*): se tira dos individuos aleatoriamente en la población y se reproduce el mejor de los dos en la nueva población. Uno retira esta este procedimiento hasta que la nueva población sea completada. Este método da buenos resultados. De todos modos, la fase de selección no crea nuevos individuos en la población. Este el rol de los operadores de cruzamiento y mutación.

El Código Netlogo 8 muestra el procedimiento de selección por torneo utilizado. El procedimiento copia la lista de topologías y fitness [`topology-list`] a un variable local e inicializa la lista ganadora [`winner-list`]. Luego define a esta última como una muestra de tamaño aleatorio de hasta la mitad de los individuos elegidos al azar de la población. Finalmente, la lista es ordenada por fitness.

Código Netlogo 8: Procedimiento de selección por torneo

```
Entrada: seed-genome (lista de conexiones de cada nodo del pool de selección), topology-list (lista de matrices de adyacencia y su pago asociado);  
Salida: winner-list (lista de los individuos elegidos por el procedimiento para apareamiento mediante la regla de torneo)  
  
to tournament  
let draftlist topology-list           ;; creo una copia de la lista de topologías  
set winner-list []                   ;; inicializo la lista de ganadores  
set draftlist shuffle topology-list   ;; desordeno completamente la copia de la lista  
;; ahora selecciono una muestra de la copia de la lista de tamaño aleatorio entre 1 y la mitad de  
;; la lista.  
  set winner-list n-of (1 + random (length draftlist / 2)) draftlist  
;; ahora ordeno la lista de mayor a menor fitness  
  set winner-list sort-by [ item (num-nodes * num-nodes) ?1 > item (num-nodes * num-nodes)  
?2] winner-list  
end
```

Una versión que ejerce todavía mayor presión genética en la selección de potenciales parejas de cruzamiento es una versión del procedimiento de selección por torneo denominada *elitismo*. La única modificación es que la selección de los individuos que van a participar en el torneo se realizan una muestra de los mejores individuos de la población precedente. Es decir, es refinar aún más el procedimiento de torneo reduciendo la diversidad genética manteniendo siempre a los mejores individuos de la población precedente en el nuevo pool de apareamiento. El Código Netlogo 9 muestra esta variante del procedimiento de selección por elitismo.

Código Netlogo 9: Procedimiento de selección por elitismo (variante de torneo)

Entrada: seed-genome (lista de conexiones de cada nodo del pool de selección), topology-list (lista de matrices de adyacencia y su pago asociado);
Salida: winner-list (lista de los individuos elegidos por el procedimiento para apareamiento mediante la regla de elitismo)

```
to elitism  
let draftlist topology-list           ;; creo una copia de lista de topologías  
set winner-list []                   ;; inicializo la lista ganadora  
;; creo la lista con una sublista de la lista de topologías ordenadas de tamaño aleatorio  
;; entre 1 (el mejor) y el tamaño de la lista.  
set winner-list sublist draftlist 0 (1 + random (length draftlist - 1))  
end
```

3.3.4.2. Operador de cruzamiento

El operador de cruzamiento permite la creación de nuevos individuos según un proceso muy simple. Permite el intercambio de información entre los cromosomas (individuos) por la forma en que se combinan. La población que resulta de la selección esta dividida en dos sub-poblaciones de tamaño $N/2$ y cada pareja formada por una miembro proveniente de cada sub-población participa de un cruzamiento con una probabilidad dada (la probabilidad de cruzamiento p_c frecuentemente es superior al 60%, aunque en nuestro caso en particular es del 100%). Si el cruzamiento tiene lugar, su localización entre la posición 1 y la posición l , en el caso de la codificación binaria, es obtenida según una ley uniforme y los dos individuos intercambian sus genes entre las dos cotas de esta localización. Puede haber, como en nuestro caso, un cruzamiento de dos puntos. Se establecen dos cotas y se intercambia el centro del cromosoma.

Dos nuevos individuos son creados por este proceso. Cada vez un individuo seleccionado a través de la reproducción no es necesariamente utilizado en la operación de cruzamiento. Este último se efectúa pero con una cierta probabilidad. Si es probabilidad es elevada la población cambiará fuertemente. Se puede imaginar que el cruzamiento tiene lugar muchas

zonas diferentes de la cadena (más que dos como en este ejemplo). El procedimiento de cruzamiento empleado se expone en el Código Netlogo 9. El cruzamiento es de dos puntos.

Código Netlogo 10. Procedimiento de cruzamiento

Entrada: ranking (lista seleccionados al pool de apareamiento mediante la regla de ruleta), winner-list (lista seleccionados al pool de apareamiento mediante la regla de torneo y elitismo);
Salida: seed-genome (lista de los individuos de la nueva población)

to make-mix-genes

```
;; Una vez seleccionados los pares a empardar toman un ítem al azar y crean el marido.
;; y la esposa determinándose aleatoriamente el punto de cruzamiento. En nuestro caso
;; es cruzamiento de dos puntos.
let k 0
let i random num-nodes * num-nodes
let v num-nodes * num-nodes - i

set mix-genes []
let husband []
let wife []
let chosen random-float 1
let swap false
while [k < i][ ifelse chosen < .5 [ set swap true
  if Selection = "Elitism" [ set husband lput item k seed-genomes husband]
  if Selection = "Roulette" [ set husband lput item k seed-genomes husband]
  if Selection = "Tournament" [ set husband lput item k item 0 winner-list husband] ]
  [
    if Selection = "Elitism" [set wife lput item k seed-genomes wife]
    if Selection = "Roulette" [set wife lput item k seed-genomes wife]
    if Selection = "Tournament" [set wife lput item k item 0 winner-list wife]
  ]
  set k k + 1
]
;;
while [k < v][ ifelse swap = true [
  if Selection = "Elitism" [set wife lput item k item random length winner-list winner-list wife]
  if Selection = "Roulette" [set wife lput item k item random length topology-list
    topology-list wife]
  if Selection = "Tournament" [set wife lput item k item 0 winner-list wife] ]
  [
    if Selection = "Elitism" [set husband lput item k item random length winner-list
    winner-list husband]
    if Selection = "Roulette" [set husband lput item k item random length topology-list
    topology-list husband]
    if Selection = "Tournament" [set husband lput item k item 0 winner-list husband]
  ]
  set k k + 1
]
while [k < num-nodes * num-nodes][ ifelse chosen < .5 [ set swap true
  if Selection = "Elitism" [ set husband lput item k seed-genomes husband]
  if Selection = "Roulette" [ set husband lput item k seed-genomes husband]
  if Selection = "Tournament" [ set husband lput item k item 0 winner-list husband] ]
  [
    if Selection = "Elitism" [set wife lput item k seed-genomes wife]
```

```

if Selection = "Roulette" [set wife lput item k seed-genomes wife]
if Selection = "Tournament" [set wife lput item k item 0 winner-list wife]
]
set k k + 1
]
End

```

3.3.4.3. Operador de mutación

El rol de este operador es modificar aleatoriamente, con una cierta probabilidad, el valor de un componente del individuo. En el caso de una codificación binaria, cada bit $a_i \in \{0,1\}$ es reemplazado con una probabilidad p_m por su complementario $\tilde{a}_i = 1 - a_i$.

El procedimiento de mutación empleado se expone en el Código Netlogo 9. Se toma el genoma ya seleccionado y cruzado (*seed-genome*) y se lo recorre en cada componente comparando la tasa de mutación con un valor aleatorio de distribución normal. Cuando éste es más bajo que la tasa de mutación se realiza el intercambio del alelo del cromosoma (si valía 0 pasar a ser 1 y viceversa).

Código Netlogo 11: Procedimiento de mutación

```

Entrada: seed-genomes (lista de individuos de la nueva población sin mutación), mutation-rate (tasa de mutación);
Salida: seed-genome (lista de los individuos de la nueva población)

to mutate
  let genome-m 0
  while [genome-m < length seed-genomes] [ ;; explorar todos los genes
    if (random-float 1000 < mutation-rate * 1000)
      ;; cambiar si la tasa de mutación es alcanzada
      [ let switch item genome-m seed-genomes
        ifelse (switch = 1)
          [ set seed-genomes (replace-item genome-m seed-genomes 0) ]
          [ set seed-genomes (replace-item genome-m seed-genomes 1) ]
      ]
    set genome-m genome-m + 1 ;; se mueve al próximo genoma
  ]
  set seed-genomes diag-zero seed-genomes
end

```

3.3.5 Principales parámetros

Los operadores del AG son guiados por un cierto número de parámetros estructurales dados. El valor de esos parámetros influye la capacidad de éxito o no y la rapidez de un AG. Vamos ahora a discutir brevemente el rol de estos parámetros en la versión simple que nosotros analizamos de los AG.

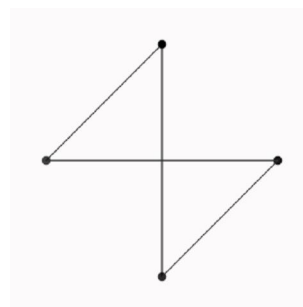
- El tamaño de la población N , y el largo de la codificación de cada individuo, l . Si N es demasiado grande los tiempos de cálculo del AG podrían ser muy importantes; si N es demasiado pequeño, puede converger rápidamente a un mal cromosoma. La importancia del tamaño es esencialmente debido a la noción de paralelismo implícito: según N sea más grande, más elevado es el número de soluciones potenciales evaluadas en paralelo por el AG.
- La probabilidad de cruzamiento p_c . Depende en general de la forma de la función de performance. Su elección es más bien heurística (también para p_m). Si es elevada, más población sufrirá cambios importantes. Los valores generalmente fluctúan en el rango de 0.5 y 0.9.
- La probabilidad de mutación p_m . Esta tasa es generalmente baja porque una tasa elevada arriesga a conducir a una solución sub-óptima en perjuicio de la óptima.
- La estructura de la función de fitness para calcular el desempeño de los individuos de la población. Según sea más complejo, más tiempo insumirá su cálculo y más difícil será para el algoritmo encontrar los cruzamientos necesarios para llegar al óptimo.

En vez de reducir p_m , otra forma de evitar que los mejores individuos sean alterados es utilizar el *elitismo*: Así, se puede elegir, por ejemplo, reproducir el idéntico al 5% de los mejores de población de cada generación, el operador de reproducción no elige ahora más que sobre el 95% del resto.

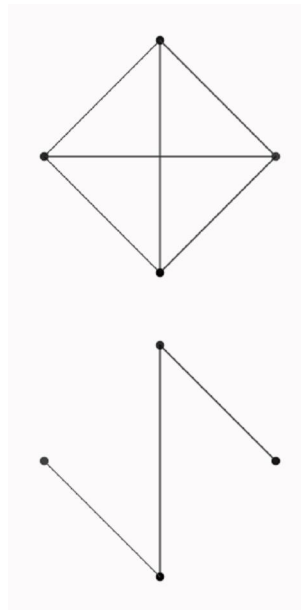
Ejemplo 2. Veamos cómo opera entonces un AG para un caso específico. Tenemos una población de 3 cromosomas y una red de 4 nodos. Los valores de las variables relevantes son para la información poseída por cada agente el valor es 1 y el costo de conexión tiene valor 10. Consideramos al grafo como dirigido, es decir que los jugadores acceden a la información conectándose a otro jugador pero no requieren de reciprocidad. El AG obtiene la solución óptima ($\Pi(g) = 4$, lo cual una red vacía). Para cada cadena, el cruzamiento, su posición y la asociada implicada son determinadas de manera aleatoria. La mutación simplemente cambia el valor del bit concerniente: de 0 a 1 o de 1 a 0. Este proceso es controlado por el tamaño de la población, el tamaño de las cadenas binarias y las probabilidades de cruzamiento y mutación.

Etapas (a). Población al momento t

1) [0011000101000100] = -41



2) [0111101101010100] = -77



3) [0100100010000010] = -32

Etapa (b) Selección por ruleta y torneo

$Fitness: \Pi(g) = I(g) - C(g)$	Número esperado de ejemplares: $\Pi(g)/Media$	Número efectivo de ejemplares (elegidos aleatoriamente)	Número efectivo de ejemplares (elegidos por torneo)
1) -41	1) 0.82	1) 0	1) 2do
2) -77	2) 1.54	2) 2	2) 3ro
3) -32	3) 0.64	3) 1	3) 1ro
Media: -50	Suma: 3	Suma: 3	Suma: 3

Etapa (c) Nueva población intermedia

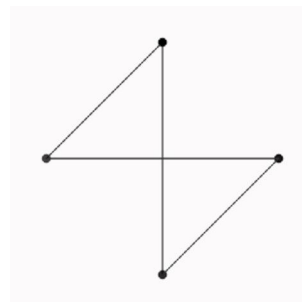
1) [0011000101000100] (= -41)

2) [0111101101010100] (= -77)

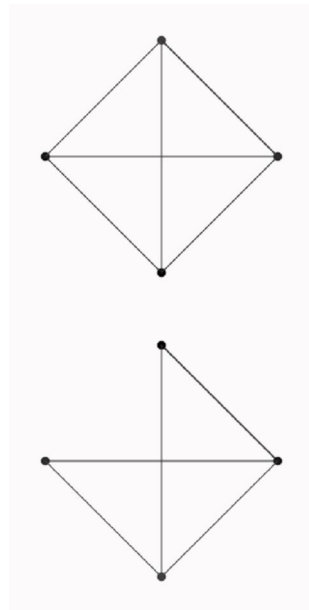
3) [0100100010000010] (= -32)

Etapa (d) Cruzamiento (selección por ruleta) al noveno bit entre el (3)-(2)

1) [0011000101000100] (= -41)



2) [0111101100000010] (= -59)



3) [0100100011010100] (= -50)

Etapa (e) Mutación de (1)

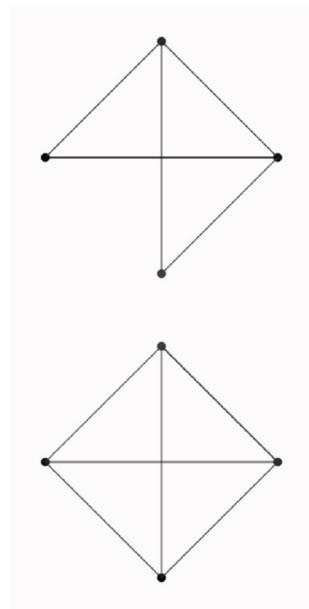
1) [0011100101000100] (= -50*)

2) [0111101100000010] (= -59)

3) [0100100011010100] (= -50)

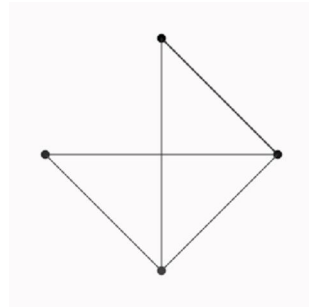
Etapa (f) Población en el momento t+1

1) [0011100101000100] (= -50)



2) [0111101100000010] (= -59)

3) [0100100011010100] (= -50)



3.4 Los algoritmos genéticos y su interpretación económica

Siendo que un AG puede ser representado como un conjunto de individuos que a través de un período de vida limitado realizan una operación económica determinan y evolucionan aprendiendo a realizar dicha transacción de manera más eficiente cada vez. Riechmann (1998) resume las principales bases teóricas asociadas a la interpretación económica de los algoritmos genéticos como instrumentos de la vertiente evolucionista de la Economía. Hay tres formas de aprendizaje por algoritmo genético (Riechmann, 2001):

- *Aprendizaje por imitación.* La selección y reproducción de estrategias (individuos) puede ser interpretados como una forma de aprendizaje por imitación: los agentes cuyas estrategias poseen un desempeño bajo (fitness relativo bajo) abandonan sus anteriores estrategias y copian la estrategia de un miembro más exitoso de la población.
- *Aprendizaje por comunicación.* El cruzamiento aleatoriamente elige dos individuos genéticos de la población y crea un nuevo individuo combinando partes de los bits de sus componentes de los cromosomas de los dos padres. Esto puede ser interpretado como una forma de aprendizaje por comunicación. Dos agentes se encuentran, hablan entre ellos acerca de sus estrategias y ello brinda la oportunidad de adaptar partes de la conducta de cada uno. Y finalmente,

- *Aprendizaje por experimentación.* La mutación aleatoriamente altera bits individuales del cromosoma por el que el AG es codificado, invirtiendo su valor. Desde una perspectiva económica, la estrategia de un agente económico puede ser ligeramente modificada alterando partes de ella. Mientras que la imitación (selección/reproducción) y la comunicación (cruzamiento) puede solamente reproducir estrategias ya en uso, la experimentación (mutación) es capaz de hallar estrategias que nunca han sido usadas antes (innovación verdadera).

El proceso de reproducción es interpretado como un proceso de aprendizaje por imitación, el proceso de cruzamiento es entendido como un proceso de aprendizaje por comunicación y, finalmente, el proceso de mutación es traducido como un proceso de aprendizaje por experimentación.

Otros trabajos puntualizan su importancia para la teoría económica, aunque desde otra perspectiva. Así, Chen (2000, 2001) describe a la programación genética como diseñada para hacer crecer (evolucionar) una población de jerarquías evolutivas de bloques constructores (subrutinas) de unidades básicas de aprendizaje e información.

Tres objetos bien definidos se precisan: *bloques constructores*, *jerarquías* y *población evolutiva de jerarquías*. Un bloque constructor es una clase de reglas de decisión definidas por algunas características específicas las cuales pueden desarrollar cierta clase de funciones. En la programación genética los bloques constructores son usualmente generados aleatoriamente por un conjunto de *primitivas*, conocidas como el conjunto de

funciones y el conjunto terminal. La primera gran cuestión es entonces: *la elección de las primitivas*.

Una vez que un conjunto de primitivas está dado, las jerarquías son derivadas por algunas reglas de producción (gramática). Dada la gramática, cualquiera jerarquía que sea *sintácticamente válida* es una especie legítima. Su apariencia y popularidad serán crucialmente dependientes del grado de ajuste (*fitness*), el cual es básicamente derivado de tres operadores genéticos: *reproducción, cruzamiento y mutación*. Las cuestiones encontradas en este estadio del análisis son dos: las *restricciones semánticas de las jerarquías derivadas* y el *uso de los operadores genéticos*.

Por otro lado, las jerarquías no son estáticas, sino dinámicamente adaptadas a un entorno que está exógenamente dado o endógenamente modificado por los agentes. La dinámica generada por la programación genética es una secuencia de conjunto de programas (árboles de búsqueda, subrutinas, ideas, estrategias). Esta secuencia puede ser interpretada como la evolución de una sociedad artificial en un todo. En otras palabras, una sociedad de agentes adaptativos tiene una relación biunívoca con una población de programas. Alternativamente, esta secuencia puede ser interpretada como la adaptación de un agente simple. En este caso, una sociedad de agentes consiste en muchas poblaciones de programas. La primera interpretación es usualmente referida a *programación genética de poblaciones simples* mientras que la segunda es denominada *programación genética de poblaciones múltiples*.

Dos enfoques principales pueden ser distinguidos en la literatura para implementar un AG en problemas económicos basados en agentes (Tabla 5). Usando el que llamaremos Enfoque I, cada agente es representado por una estrategia simple en la población genética. El segundo enfoque permite a un agente seleccionar una estrategia de una población más grande de estrategias. En el Enfoque I el valor del tamaño del parámetro poblacional del algoritmo genético (esto es, el número de estrategias del pool genético) está directamente determinado por el modelo económico. En el Enfoque II éste no es el caso. Usando el Enfoque II (Axelrod, 1998), tenemos que fijar el valor para el tamaño de la población de modo tal que los resultados económicos sean robustos a cambios en los valores de los parámetros (tanto económicos como genéticos).

Tabla 5. Enfoques usados en AG en la literatura económica

<i>Enfoque I</i>	<i>Enfoque II</i>
Cromosoma = Agente	Cromosoma = Estrategia
Tamaño de la población determinado por el número de agentes económicos	Tamaño de la población determinado por las propiedades de aprendizaje del propio algoritmo evolutivo

El AG representa una técnica más cuantitativa y versátil para modelar la conducta del agente cuando existe racionalidad acotada y numerosas interacciones secuenciales son requeridas para registrar un patrón de aprendizaje agentes con información imperfecta.

3.5 La estabilidad en el aprendizaje por algoritmos genéticos

La estabilidad en la literatura principal económica describe una situación en la cual –luego de algún estado de dinámica transicional- un estado es establecido en el cual las decisiones de los agentes económicos dejan de cambiar (estabilidad asintótica), cambia dentro de un espacio restringido alternativas (estabilidad de Ljapunov) o cambia de alguna manera regular (estabilidad cíclica). Ésta son nociones de macro-estabilidad. Los aprendizajes mínimos

genéticos demuestran varias combinaciones de estabilidad a nivel micro y macro. Algunos ejemplos son los siguientes:

- Los datos agregados permanecen inmutables en el tiempo (estabilidad asintótica) y la conducta individual es idéntica para todos los agentes económicos.
- Los datos agregados permanecen inmutables en el tiempo (estabilidad asintótica) mientras que la conducta individual es heterogénea y cambia de tiempo en tiempo.
- Los datos agregados provienen del conjunto finito de números (estabilidad de Ljapunov) entre la conducta del individual es heterogénea y cambia de modo tal que un número finito de patrones de conducta social son observados.
- Los datos agregados cambian regularmente en ciclos (estabilidad cíclica) mientras que diferentes agentes se comportan diferentemente en el mismo período y cada agente cambia su conducta de tiempo en tiempo.

Básicamente en el presente trabajo la estabilidad es observada en gran parte de las simulaciones realizadas. Los valores de las principales variables analizadas (nivel de beneficios agregados y densidad de red) alcanzan estados estacionarios estables alrededor de valores que en gran parte de las ocasiones coinciden con el valor teórico de los modelos formales de formación de red.

Figura 5. Estabilidad de densidad de red y beneficios en una red de 10 nodos

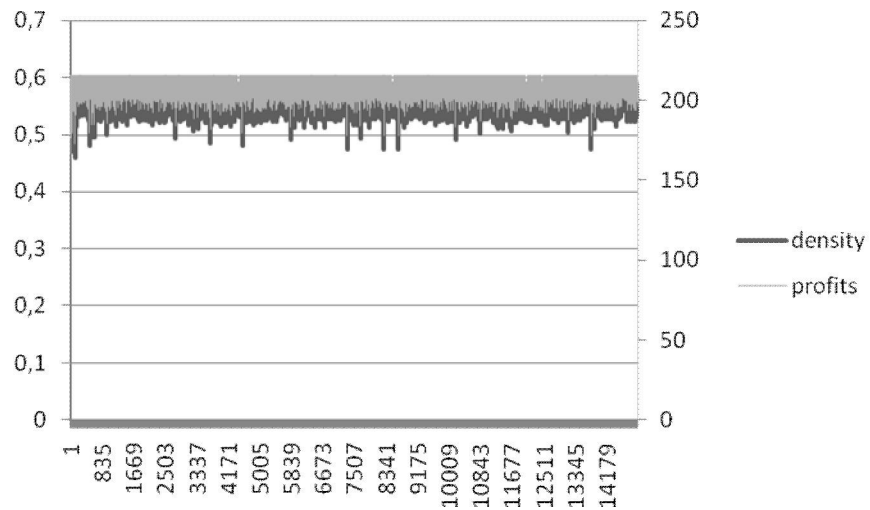
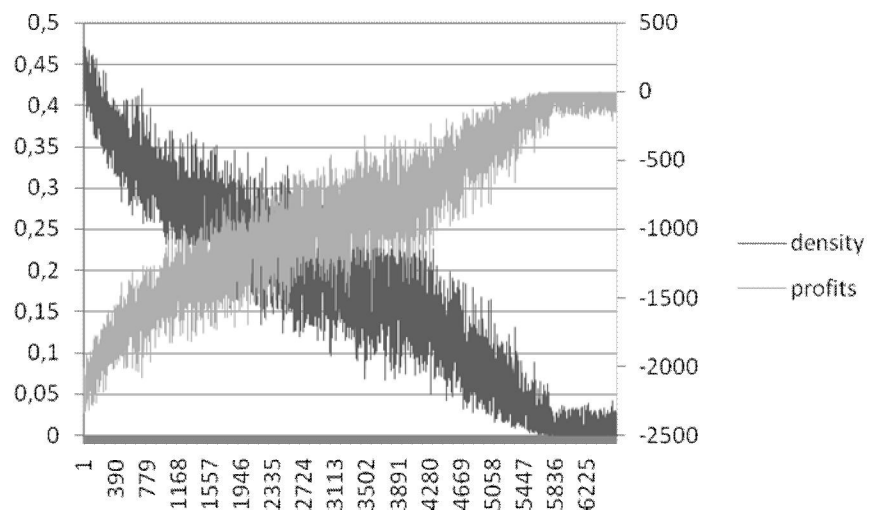


Figura 6. Estabilidad de densidad de red y beneficios en una red de 30 nodos



A continuación se sigue con el esquema y las definiciones. En términos del campo de aplicación de los algoritmos genéticos, es decir la Economía, la interpretación de cómo interpretar los resultados merece una explicación particular.

Capítulo 4. Análisis de datos

4.1 Introducción

El presente capítulo exhibirá los resultados de las simulaciones realizadas con el código presentado a lo largo del capítulo 3. Las simulaciones son realizadas a fin de observar que grado de aproximación tiene el algoritmo genético (AG) de formación de redes respecto a las hipótesis planteadas por los modelos teóricos revisados en el capítulo 2. En ese sentido, se espera *ex ante* que el código genere resultados similares a los propuestos por los modelos formales y caso que sean aproximados (o no tan aproximados) se intentará deducir que variables han afectado esta tarea.

En grandes líneas, se observa en primer lugar que el AG diseñado se comporta adecuadamente aunque en muchos casos el nivel de convergencia a la solución requiere de mayor cantidad de generaciones de las utilizadas en las simulaciones. En ese sentido la tendencia tiene el signo de la pendiente de evolución de las variables de desempeño es el esperado pero al finalizar las corridas todavía dicho valor no se ha alcanzado.

En segundo lugar, la escasez de generaciones observable necesarias para la convergencia se encuentra altamente relacionada con la combinatoriedad del problema enfrentado. Las

potenciales soluciones a investigar en el caso de las redes de 10 nodos es de $10^{29} = 1.000.000.000.000.000.000$, es decir un trillón de posibilidades a analizar, y para el caso más extenso analizado de 30 nodos, las alternativas a evaluar son de $30^{29} = 4,71e+110$, siendo que en muchos casos se ha alcanzado el valor teórico con menos de 15000 generaciones.

En tercer lugar, el AG detecta bien los casos extremos. Cuando el costo relativo de la información es muy alto (un décimo del valor del costo de conexión, por ejemplo) el algoritmo converge rápidamente a la topología de red vacía. Lo mismo ocurre para el caso de costo relativo de la información muy bajo (diez veces superior al costo de conexión) donde rápidamente se converge a la topología de red completa. Sin embargo, las topologías intermedias más estilizadas (red de estrella centralmente sostenida) sólo raramente han sido encontradas. En los casos intermedios el AG converge a niveles de densidad de red intermedio (valores de 0.45 a 0.6 en promedio). Este problema ha sido encontrado en la literatura experimental de formación de redes. Sujetos sometidos a juegos experimentales de formación de redes encuentran fácilmente topologías extremas pero fallan en encontrar las topologías intermedias más refinadas de acuerdo a la deducción formal del juego. La red de estrella centralmente sostenida es sólo accidentalmente encontrada en juegos experimentales con humanos (Berninghaus et al., 2004; Berninghaus et al., 2007; Falk y Kosfeld, 2003).

4.2 Patrones de datos para cada hipótesis de la investigación

Iniciamos esta sección analizando las hipótesis desprendidas de los modelos de Bala y Goyal (2000) y Larrosa y Tohmé (2003b, 2005), los cuales son modelos de formación de redes básicos. Luego se seguirá con el análisis de los modelos de colaboración en redes de Goyal y Joshi (2003) y Billand y Bravard (2004).

Cada simulación presenta inicialmente las hipótesis planteadas por el modelo teórico respectivo con sus variedades presentadas en el Capítulo 2. Luego se presentan las parametrizaciones utilizadas para todas las corridas. Las simulaciones se dividen en redes de tamaño predeterminados. Por problemas de tiempo de ejecución del código, sólo se analizarán redes de 5, 10 y 20 nodos con 15000 generaciones en cada caso. Al iniciar cada simulación se presenta una pequeña tabla con los valores críticos que justifican los valores adoptados para observar el surgimiento de una topología.

Cada simulación fue realizada empleando los tres procesos de selección expuestos en el Capítulo 3: selección por torneo (Tournament), selección por elitismo (Elitism) y selección por ruleta (Roulette). Asimismo, cada simulación fue repetido con tres valores crecientes de tasas de mutación: 0, 0.05 y 0.1.

4.2.1 Modelo de Bala y Goyal

Para el modelo de comunicación monodireccional con acceso a información indirecta con costos de conexión directos con tasa de decaimiento en la información δ (BG2000: 1194-1202) recordamos que requerimientos

- VII. *Dada la estructura de pagos (2.3), una red con comunicación monodireccional y presencia de pérdida de calidad de información tiene una estructura eficiente si :*
- a. *Es una red completa cuando $0 < c < 2[\delta - \delta^2]$ (H1),*
 - b. *Es una red de estrella cuando $2[\delta - \delta^2] < c < 2\delta + (n-2)\delta^2$ (H2), y*
 - c. *Es una red vacía cuando $c > 2\delta + (n-2)\delta^2$ (H3) (Proposición 4.1 , BB2000)*

Para el modelo de comunicación bidireccional con acceso a información indirecta con costos de conexión directos (BG2000: 1202-1209)

- VIII. *Dada la estructura de pagos (2.3), una red con comunicación bidireccional y presencia de pérdida de calidad de información tiene una estructura eficiente si :*
- a. *Es una red completa cuando $c < \delta - \delta^2$ (H4),*
 - b. *Es una red de estrella cuando $\delta - \delta^2 < c < \delta + [(n-2)/2]\delta^2$ (H5), y*
 - c. *Es una red vacía cuando $c > \delta + [(n-2)/2]\delta^2$ (H6) (Proposición 4.1 , BB2000)*

La Tabla 6 muestra los parámetros del AG que representa al modelo de Bala y Goyal (2000).

Tabla 6. Parámetros del AG

Número de agentes	5	10	20
Generaciones por corrida	15000		
Corridas por caso	30		
Tamaño de la población	200		
Flujo de información	Monodireccional (grafo dirigido)	Bidireccional (grafo no dirigido)	
Función de desempeño (fitness)	Función de beneficios con costo directo e información directa con información	Función de beneficios con costo directo e información directa con información	
Tipo de selección	Tournament	Elitism	Roulette
Estilo de cruzamiento	Cruzamiento de 2 puntos		
Tasa de cruzamiento	1		
Tasa de mutación	0	0.05	0.1
Tasa de decaimiento (δ)	0.5		
Valor relativo de la información (valor info/costo)	0.35	1	8

4.2.1.1 Red de 5 nodos

Primeramente se estiman los valores críticos de las H1 a H3 para el caso de 5 nodos. Dada la forma en que se modelizó la relación entre costo e ingreso por información, el `ratio-info-edge` representa la inversa (dado que el valor de propia información fue estandarizado a 1) del costo crítico expuesto.

Tabla 7. Valores críticos para red de 5 nodos en caso monodireccional de Bala y Goyal (2000)

Topología	Valor de c	Valor <code>ratio-info-edge</code>
Red completa	$2[\delta - \delta^2] = 0.5$	2
Red de estrella		
Red vacía	$2\delta + (n-2)\delta^2 = 1.75$	0.57

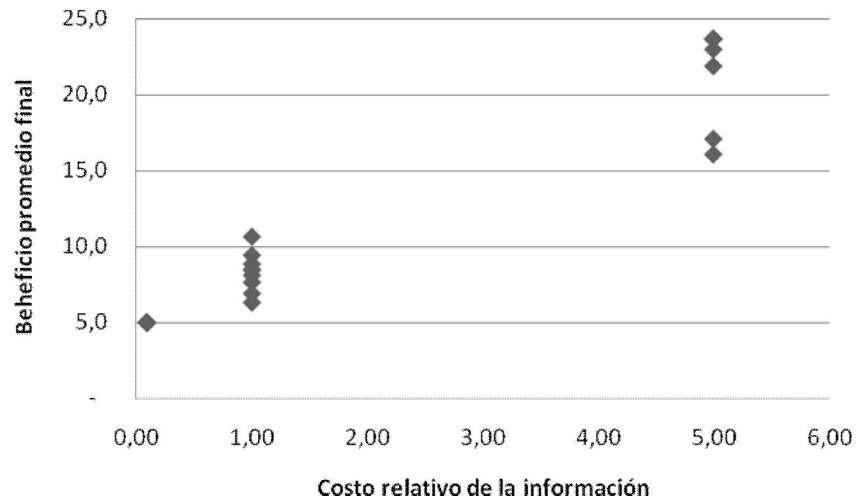
La Tabla 8 muestra los resultados de 30 corridas con la parametrización descrita en la Tabla 6. Significativamente el ratio de acierto es muy alto para valores bajos y altos de la simulación.

Tabla 8. Red de 5 nodos, comunicación monodireccional con decaimiento ($\delta = 0.5$)

ratio-info-edge	Selección	Tasa de Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.09	Tournament	0	5	0	5.0	0.00	1.0	67
0.09	Elitism	0	5	0	5.0	0.00	1.0	157
0.09	Roulette	0	5	0	5.0	0.00	1.0	1565
0.09	Tournament	0.05	5	0	5.0	0.00	1.0	108
0.09	Elitism	0.05	5	0	5.0	0.00	1.0	85
0.09	Roulette	0.05	5	0	5.0	0.00	1.0	1338
0.09	Tournament	0.1	5	0	5.0	0.00	1.0	132
0.09	Elitism	0.1	5	0	5.0	0.00	1.0	156
0.09	Roulette	0.1	5	0	5.0	0.00	1.0	2173
1	Tournament	0	5.8	0.20	8.9	0.43	0.0	15000
1	Elitism	0	5.8	0.20	7.7	0.45	0.0	13610
1	Roulette	0	5.8	0.20	6.9	0.41	0.0	15000
1	Tournament	0.05	5.8	0.20	10.7	0.57	0.0	15000
1	Elitism	0.05	5.8	0.20	9.4	0.59	0.0	15000
1	Roulette	0.05	5.8	0.20	6.3	0.48	0.0	15000
1	Tournament	0.1	5.8	0.20	8.1	0.56	0.0	15000
1	Elitism	0.1	5.8	0.20	8.4	0.62	0.0	15000
1	Roulette	0.1	5.8	0.20	8.5	0.52	0.0	15000
5	Tournament	0	23.7	1	16.1	0.57	1.0	15000
5	Elitism	0	23.7	1	23.7	1.00	1.0	2269
5	Roulette	0	23.7	1	17.1	0.60	0.0	15000
5	Tournament	0.05	23.7	1	23.6	0.97	0.9	3363
5	Elitism	0.05	23.7	1	23.0	0.91	0.6	6816
5	Roulette	0.05	23.7	1	23.7	1.00	1.0	6735
5	Tournament	0.1	23.7	1	23.7	1.00	1.0	795
5	Elitism	0.1	23.7	1	21.9	0.91	0.7	4795
5	Roulette	0.1	23.7	1	23.7	1.00	1.0	6383

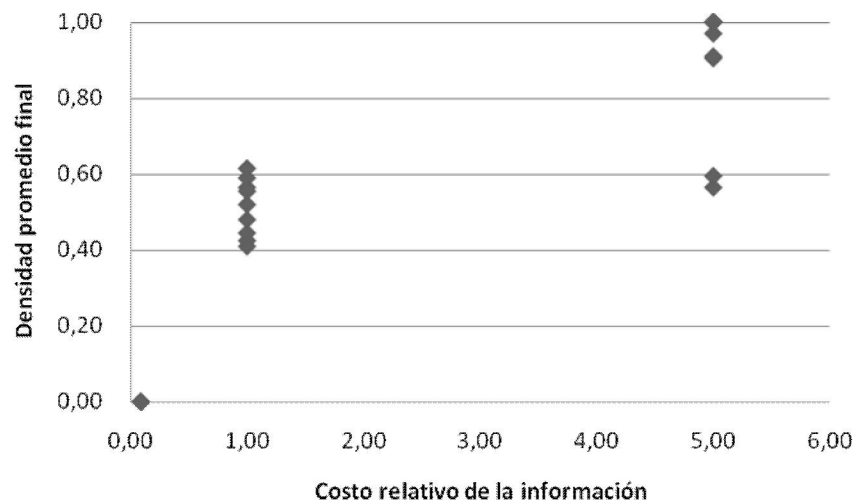
El Gráfico 1 muestra la relación entre el beneficio promedio de todas las corridas y el costo relativo de la información (representado por el parámetro *ratio-info-edge*). Se distingue claramente un agrupamiento único en beneficios de valor 5 y mayor variabilidad a medida que disminuye el costo relativo de la información).

Gráfico 1. Beneficio promedio final y costo relativo de la información (5 nodos)



El Gráfico 2 muestra una relación similar a la anterior para la densidad de red y el costo relativo de la información.

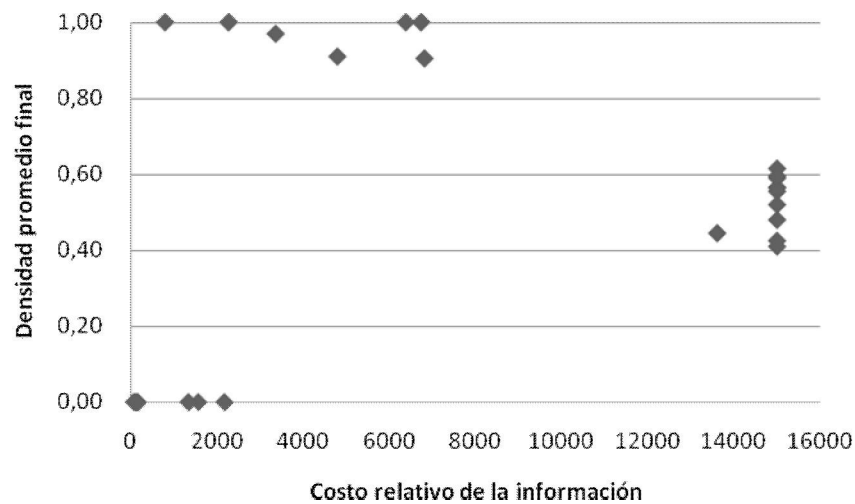
Gráfico 2. Densidad de red final y costo relativo de la información (5 nodos)



Finalmente el Gráfico 3 muestra la relación entre la densidad de red (como identificador de la topología de red predominante) y el número de generaciones hasta alcanzar el valor

óptimo teórico (en caso de no alcanzarlo el AG simplemente recorrió las 15000 generaciones de cada corrida).

Gráfico 3. Densidad de red final y generaciones (5 nodos)



Se observa que existen muchas corridas que se agrupan sobre las 15000 generaciones, correspondientes a un costo relativo de la información de valor 1. Allí no se ha definido una estructura de red clara y con ello queda estable en un valor intermedio.

Ahora examinemos el caso bidireccional (grafos no dirigidos) de Bala y Goyal (2000). Primero se observan nuevos valores críticos en la Tabla 9.

Tabla 9. Valores críticos para red de 5 nodos en caso bidireccional de Bala y Goyal (2000)

Topología	Valor de c	Valor <i>ratio-info-edge</i>
Red completa	$\delta - \delta^2 = 0.25$	4
Red de estrella		
Red vacía	$\delta + [(n-2)/2]\delta^2 = 0.875$	1.143

Los resultados de las corridas se presentan en la Tabla 10.

Tabla 10. Red de 5 nodos, comunicación bidireccional con decaimiento ($\delta = 0.5$)

ratio-info-edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.35	Tournament	0	5	0	3.7	0.1	0.81	9001
0.35	Elitism	0	5	0	5.0	0	1	362
0.35	Roulette	0	5	0	5.0	0	1	311
0.35	Tournament	0.05	5	0	5.0	0	1	298
0.35	Elitism	0.05	5	0	5.0	0	1	126
0.35	Roulette	0.05	5	0	5.0	0	1	273
0.35	Tournament	0.1	5	0	5.0	0	1	100
0.35	Elitism	0.1	5	0	5.0	0	1	338
0.35	Roulette	0.1	5	0	5.0	0	1	266
2	Tournament	0	7.8	0.20	19.8	0.9	0	9010
2	Elitism	0	7.8	0.20	22.7	1	0	435
2	Roulette	0	7.8	0.20	22.7	1	0	223
2	Tournament	0.05	7.8	0.20	22.7	1	0	550
2	Elitism	0.05	7.8	0.20	22.7	1	0	395
2	Roulette	0.05	7.8	0.20	22.7	1	0	138
2	Tournament	0.1	7.8	0.20	22.7	1	0	223
2	Elitism	0.1	7.8	0.20	15.6	0.6	0	136
2	Roulette	0.1	7.8	0.20	15.6	0.6	0	161
5	Tournament	0	25.7	1	18.1	0.6	0	9001
5	Elitism	0	25.7	1	25.7	1	1	347
5	Roulette	0	25.7	1	25.7	1	1	159
5	Tournament	0.05	25.7	1	25.7	1	1	130
5	Elitism	0.05	25.7	1	25.7	1	1	204
5	Roulette	0.05	25.7	1	25.7	1	1	146
5	Tournament	0.1	25.7	1	25.7	1	1	291
5	Elitism	0.1	25.7	1	25.7	1	1	215
5	Roulette	0.1	25.7	1	25.7	1	1	396

Los Gráfico 4 y Gráfico 5 muestran claramente cómo antes valores bajos del costo relativo de la información los beneficios y la densidad de red se mueven paralelamente. Obsérvese que en el caso de la densidad de red, para valores intermedios del costo relativo de la información que debieran conducir a topologías intermedias terminan conduciendo en gran parte de los casos a la red completa. Evidentemente el AG no capturó la estructura de red de estrella centralmente sostenida.

Gráfico 4. Beneficio promedio final y costo relativo de la información (5 nodos)

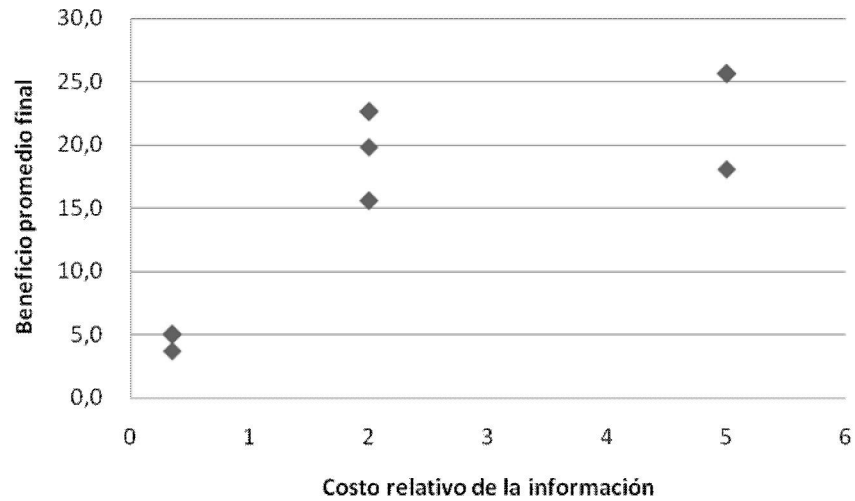
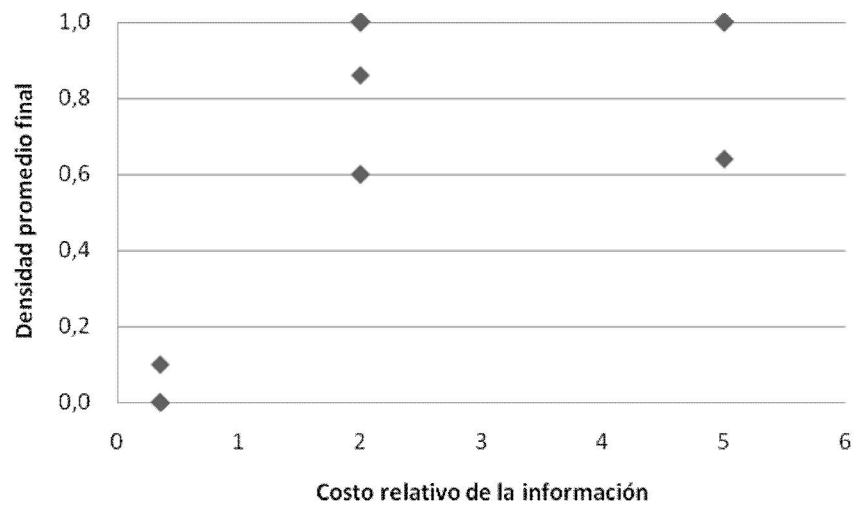
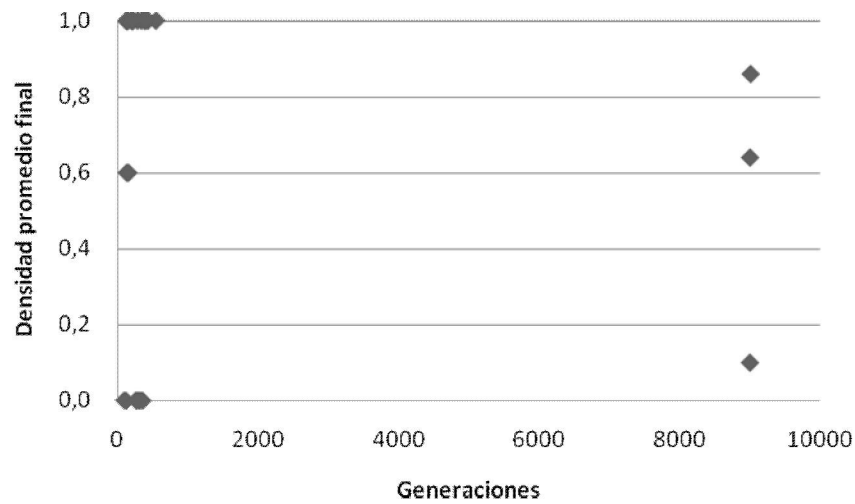


Gráfico 5. Densidad de red final y costo relativo de la información (5 nodos)



Finalmente el Gráfico 6 muestra que las diferentes corridas han utilizado plenamente el número de generaciones máximo, dado que ninguna corrida alcanzó el máximo de 15000 generaciones.

Gráfico 6. Densidad de red final y costo relativo de la información (5 nodos)



4.2.1.2 Red de 10 nodos

Mostramos inicialmente los valores críticos para el modelo de comunicación monodireccional.

Tabla 11. Valores críticos para red de 10 nodos en caso monodireccional de Bala y Goyal (2000)

Topología	Valor de c	Valor <i>ratio-info-edge</i>
Red completa	$2[\delta - \delta^2] = 0.5$	2
Red de estrella		
Red vacía	$2\delta + (n-2)\delta^2 = 3$	0.33

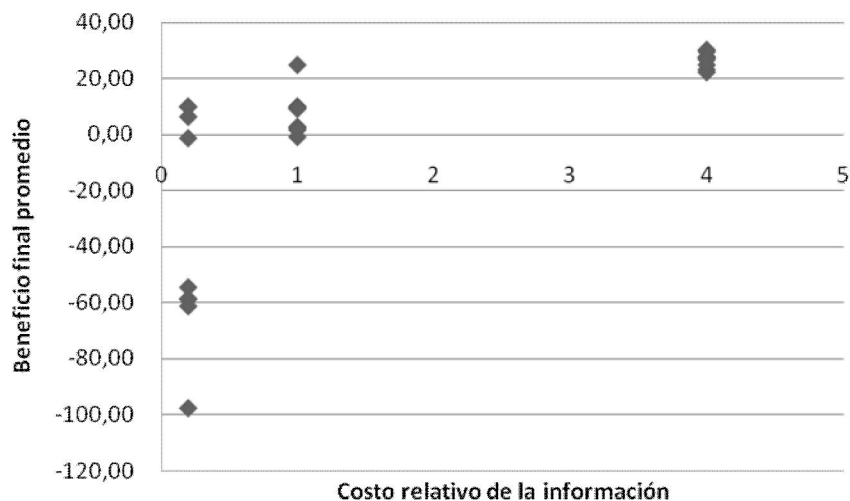
La Tabla 12 muestra los resultados de 30 corridas con la parametrización descrita en la Tabla 6. Significativamente el ratio de acierto es mucho más variable que en el caso precedente.

Tabla 12. Red de 10 nodos, comunicación monodireccional con decaimiento ($\delta = 0.5$)

ratio-info-edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.09	Tournament	0	10	0	10.0	0.0	1.0	335.7
0.09	Elitism	0	10	0	10.0	0.0	1.0	1156.3
0.09	Roulette	0	10	0	-151.2	0.2	0.0	15000.0
0.09	Tournament	0.05	10	0	10.0	0.0	1.0	3857.1
0.09	Elitism	0.05	10	0	10.0	0.0	1.0	4403.9
0.09	Roulette	0.05	10	0	-189.8	0.2	0.0	15000.0
0.09	Tournament	0.1	10	0	-121.3	0.1	0.0	15000.0
0.09	Elitism	0.1	10	0	-133.5	0.2	0.0	15000.0
0.09	Roulette	0.1	10	0	-231.9	0.3	0.0	15000.0
1	Tournament	0	12.2	0.10	21.8	0.3	0.0	15000.0
1	Elitism	0	12.2	0.10	18.0	0.4	0.0	15000.0
1	Roulette	0	12.2	0.10	18.6	0.4	0.0	15000.0
1	Tournament	0.05	12.2	0.10	21.7	0.5	0.0	15000.0
1	Elitism	0.05	12.2	0.10	20.1	0.5	0.0	15000.0
1	Roulette	0.05	12.2	0.10	15.6	0.5	0.0	15000.0
1	Tournament	0.1	12.2	0.10	20.6	0.5	0.0	15000.0
1	Elitism	0.1	12.2	0.10	19.2	0.5	0.0	15000.0
1	Roulette	0.1	12.2	0.10	16.7	0.5	0.0	15000.0
4	Tournament	0	42.5	1	53.3	0.6	0.0	15000.0
4	Elitism	0	42.5	1	63.2	0.6	0.0	15000.0
4	Roulette	0	42.5	1	50.0	0.5	0.0	15000.0
4	Tournament	0.05	42.5	1	76.4	0.8	0.0	15000.0
4	Elitism	0.05	42.5	1	76.1	0.8	1.0	14749.9
4	Roulette	0.05	42.5	1	59.8	0.6	0.0	15000.0
4	Tournament	0.1	42.5	1	67.1	0.7	0.0	15000.0
4	Elitism	0.1	42.5	1	76.3	0.7	0.0	15000.0
4	Roulette	0.1	42.5	1	48.5	0.6	0.0	15000.0

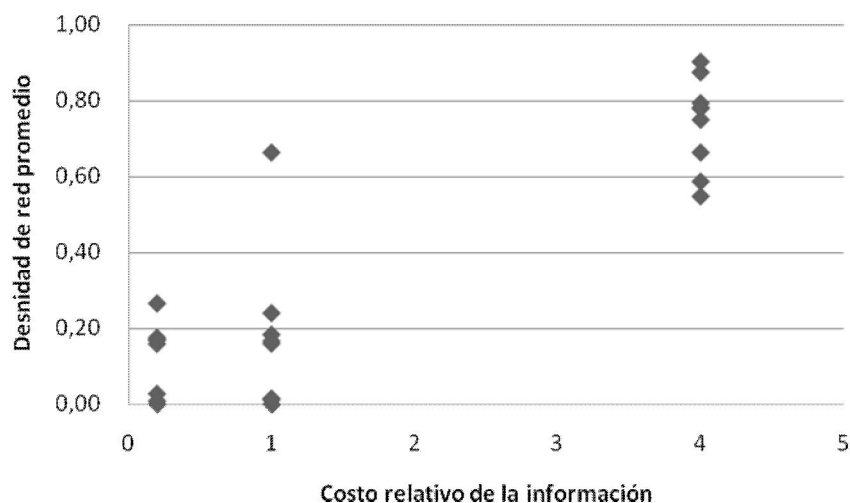
El Gráfico 7 muestra la relación entre el beneficio promedio de todas las corridas y el costo relativo de la información. Se distingue claramente una evolución desde valores negativos de performance hacia positivos cuando el costo relativo es mayor.

Gráfico 7. Beneficio promedio final y costo relativo de la información (10 nodos)



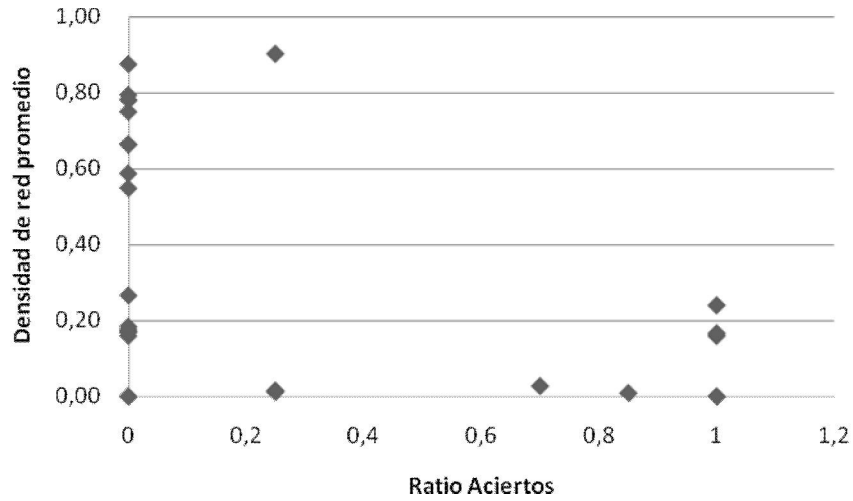
El Gráfico 8 muestra la relación entre la densidad promedio final de todas las corridas y el costo relativo de la información. Se distingue claramente una evolución desde valores cercanos a cero aunque en los valores intermedios de costo relativo de la información persiste un rango de valores previamente observado en costos mayores de acceso a la información.

Gráfico 8. Densidad de red final y costo relativo de la información (10 nodos)



El Gráfico 9, finalmente, muestra una poco clara relación entre la densidad y el ratio de aciertos.

Gráfico 9. Densidad de red promedio y ratio de aciertos (10 nodos)



Ahora analicemos el caso de **comunicación bidireccional**. Primeramente se publican los valores críticos de la red de 10 nodos en la Tabla 13.

Tabla 13. Valores críticos para red de 10 nodos en caso bidireccional de Bala y Goyal (2000)

Topología	Valor de c	Valor <i>ratio-info-edge</i>
Red completa	$\delta - \delta^2 = 0.25$	4
Red de estrella		
Red vacía	$\delta + [(n-2)/2]\delta^2 = 1.5$	0.667

A continuación en la Tabla 14 se presentan los resultados de cada simulación.

Tabla 14. Red de 10 nodos, comunicación bidireccional con decaimiento ($\delta = 0.5$)

ratio-info-edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.35	Tournament	0	10	0	-25.84	0.43	0	10000
0.35	Elitism	0	10	0	5.08	0.06	0.85	5607.6
0.35	Roulette	0	10	0	8.24	0.02	0.95	4986.4
0.35	Tournament	0.05	10	0	10.00	0	1	3299.05
0.35	Elitism	0.05	10	0	-4.86	0.18	0.45	7869.65
0.35	Roulette	0.05	10	0	-6.71	0.20	0.35	9283.5
0.35	Tournament	0.1	10	0	9.35	0.01	0.95	7755.4
0.35	Elitism	0.1	10	0	-20.27	0.36	0	9992.3
0.35	Roulette	0.1	10	0	-15.02	0.30	0.05	9990
2	Tournament	0	16.7	0.10	22.25	0.54	1	10000
2	Elitism	0	16.7	0.10	30.73	0.92	0.25	6549.9
2	Roulette	0	16.7	0.10	32.50	1.00	0	5332.45
2	Tournament	0.05	16.7	0.10	32.50	1.00	0	3912
2	Elitism	0.05	16.7	0.10	28.50	0.82	0.5	8274.6
2	Roulette	0.05	16.7	0.10	33.12	0.64	0.05	9886.25
2	Tournament	0.1	16.7	0.10	32.00	0.98	0.15	7012.2
2	Elitism	0.1	16.7	0.10	25.85	0.70	0.85	9477.9
2	Roulette	0.1	16.7	0.10	24.88	0.66	1	10000
5	Tournament	0	46	1	30.16	0.56	1	10000
5	Elitism	0	46	1	41.76	0.88	0.7	6808.4
5	Roulette	0	46	1	46.00	1.00	1	6089.8
5	Tournament	0.05	46	1	46.00	1.00	1	3909.35
5	Elitism	0.05	46	1	37.20	0.76	0.3	8720.75
5	Roulette	0.05	46	1	41.52	0.88	0.65	8733.2
5	Tournament	0.1	46	1	45.36	0.98	0.8	6950.8
5	Elitism	0.1	46	1	32.88	0.64	0	10000
5	Roulette	0.1	46	1	33.12	0.64	0.05	9886.25

Los Gráfico 10 y Gráfico 11 muestran nuevamente un paralelismo entre la densidad de red y los beneficios promedios, nuevamente asociados a una tendencia general en los valores intermedios a no detectar la estructura de estrella y seguir evolucionando hacia una estructura de red completa.

Gráfico 10. Beneficio promedio y costo relativo de la información (10 nodos)

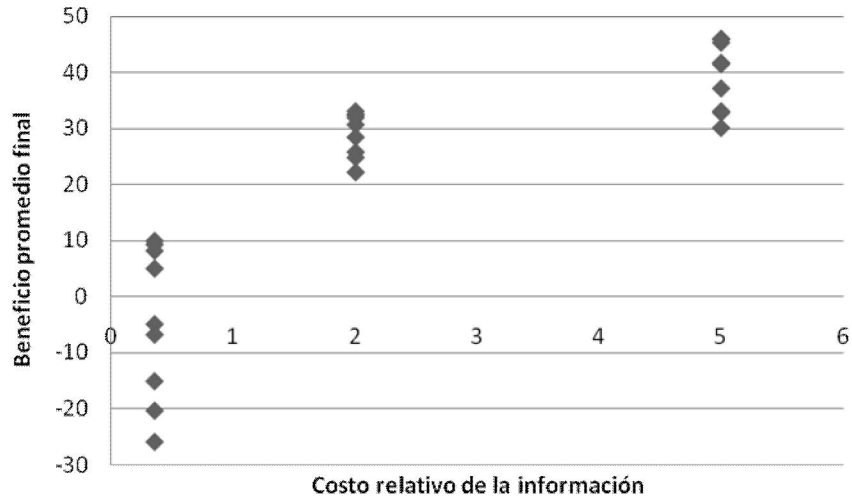
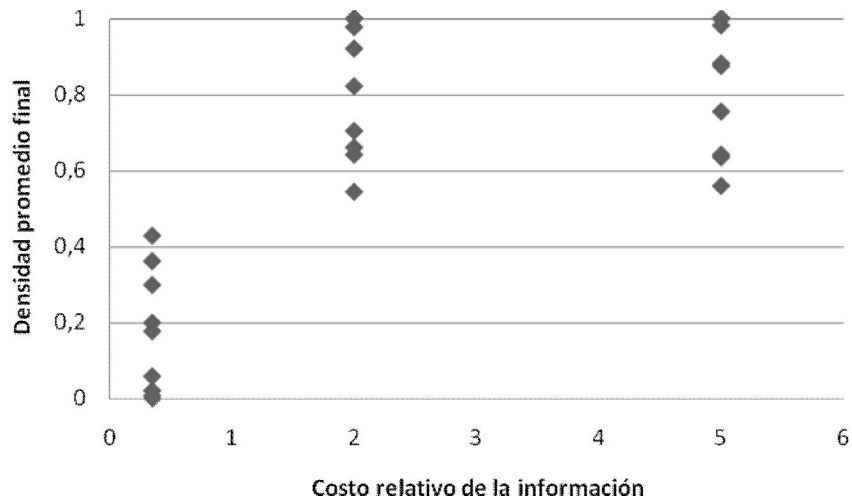
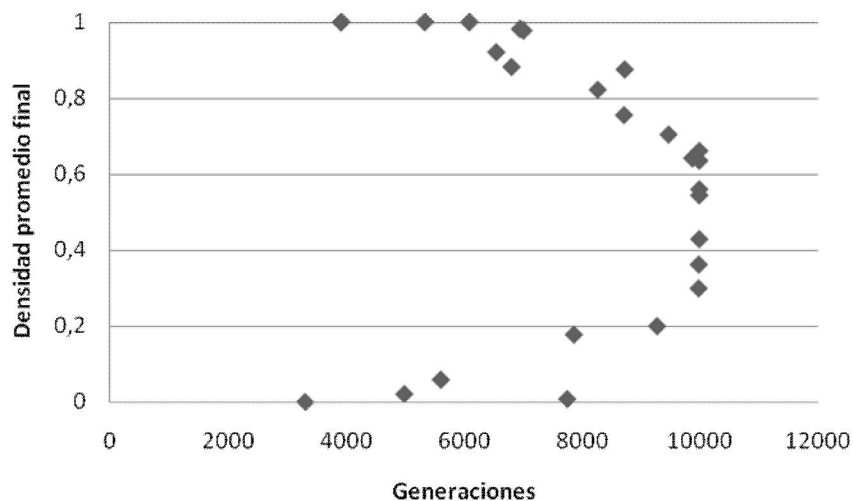


Gráfico 11. Densidad de red final y costo relativo de la información (10 nodos)



El número de generaciones presenta una relación no lineal con respecto a la densidad de red (Gráfico 12). Nuevamente, aquí se observa que los resultados de equilibrio de red vacía y red completa son obtenidos en un número promedio relativamente corto de generaciones mientras que para los resultados intermedios la conducta social estable queda cortada por alcanzar el máximo de generaciones de la corrida.

Gráfico 12. Densidad de red y generaciones promedio (10 nodos)



4.2.1.3 Red de 20 nodos

Presentamos en la Tabla 15 los valores críticos para el presente caso.

Tabla 15. Valores críticos para red de 20 nodos en caso monodireccional de Bala y Goyal (2000)

Topología	Valor de c	Valor <i>ratio-info-edge</i>
Red completa	$2[\delta - \delta^2] = 0.5$	2
Red de estrella		
Red vacía	$2\delta + (n-2)\delta^2 = 5.5$	0.18

La Tabla 16 presenta el resultado de las corridas antes diferentes costos de oportunidad de la información, procesos de selección y tasas de mutación.

Tabla 16. Comunicación monodireccional, tasa de decaimiento 0.5

ratio-info-edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.09	Tournament	0	20	0	20	0.00	1	892.4
0.09	Elitism	0	20	0	20	0.00	1	3603.3
0.09	Roulette	0	20	0	-1,138	0.30	0	15000
0.09	Tournament	0.05	20	0	-661	0.17	0.23	15000
0.09	Elitism	0.05	20	0	-697	0.18	0	15000
0.09	Roulette	0.05	20	0	-1,220	0.32	0	15000
0.09	Tournament	0.1	20	0	-1,111	0.29	0	15000
0.09	Elitism	0.1	20	0	-1,187	0.31	0	15000
0.09	Roulette	0.1	20	0	-1,455	0.38	0	15000
1	Tournament	0	24.8	0.05	21	0.32	0	15000
1	Elitism	0	24.8	0.05	17	0.40	0	15000
1	Roulette	0	24.8	0.05	19	0.43	0	15000
1	Tournament	0.05	24.8	0.05	26	0.48	0	15000
1	Elitism	0.05	24.8	0.05	23	0.46	0	15000
1	Roulette	0.05	24.8	0.05	13	0.51	0	15000
1	Tournament	0.1	24.8	0.05	23	0.52	0	15000
1	Elitism	0.1	24.8	0.05	17	0.49	0	15000
1	Roulette	0.1	24.8	0.05	12	0.52	0	15000
5	Tournament	0	360	1	170	0.48	0	15000
5	Elitism	0	360	1	183	0.50	0	15000
5	Roulette	0	360	1	177	0.47	0	15000
5	Tournament	0.05	360	1	236	0.62	0	15000
5	Elitism	0.05	360	1	260	0.62	0	15000
5	Roulette	0.05	360	1	183	0.55	0	15000
5	Tournament	0.1	360	1	196	0.57	0	15000
5	Elitism	0.1	360	1	157	0.55	0	15000
5	Roulette	0.1	360	1	149	0.53	0	15000

El Gráfico 13 muestra la relación entre el beneficio promedio de cada serie y el costo relativo de la información. Obsérvese que la falta de convergencia (escasez de generaciones) ha hecho que muchos resultados promedio de beneficios sean negativos (no alcancen el beneficio de 20) mientras que

Gráfico 13. Costo relativo de la información y beneficio promedio (20 nodos)

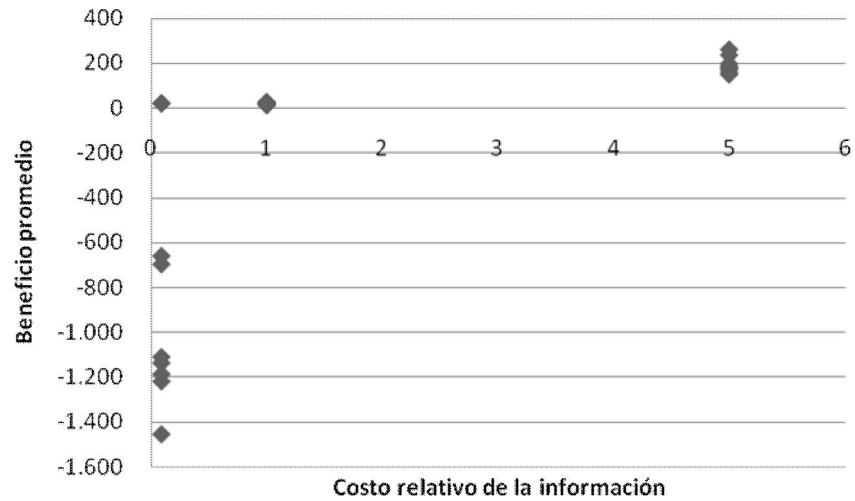
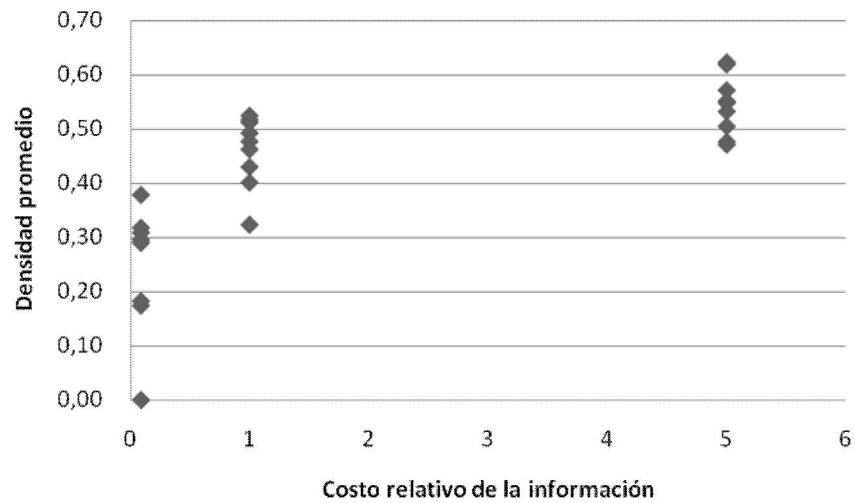


Gráfico 14. Densidad de red y costo relativo de la información (20 nodos)



Ahora analicemos el caso de **comunicación bidireccional**. Primeramente se publican los valores críticos de la red de 20 nodos en la Tabla 17.

Tabla 17. Valores críticos para red de 20 nodos en caso bidireccional de Bala y Goyal (2000)

Topología	Valor de c	Valor <i>ratio-info-edge</i>
Red completa	$\delta - \delta^2 = 0.25$	4
Red de estrella		
Red vacía	$\delta + (n-2)\delta^2/2 = 2.75$	0.364

La Tabla 18 nos muestra los resultados de las simulaciones para el caso bidireccional.

Tabla 18. Comunicación bidireccional, tasa de decaimiento 0.5

ratio-info-edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.09	Tournament	0	20	0	-793.8	0.45	0	15000
0.09	Elitism	0	20	0	-661.5	0.38	0.1	14402.9
0.09	Roulette	0	20	0	20.0	0.00	1	8904.4
0.09	Tournament	0.05	20	0	-743.4	0.43	0	15000
0.09	Elitism	0.05	20	0	-741.2	0.42	0	15000
0.09	Roulette	0.05	20	0	-275.9	0.16	0	15000
0.09	Tournament	0.1	20	0	-805.1	0.46	0	15000
0.09	Elitism	0.1	20	0	-787.6	0.45	0	15000
0.09	Roulette	0.1	20	0	-532.6	0.31	0	15000
1	Tournament	0	24.8	0.05	101.9	0.51	0	15000
1	Elitism	0	24.8	0.05	99.7	0.52	0	15000
1	Roulette	0	24.8	0.05	95.9	0.50	0	15000
1	Tournament	0.05	24.8	0.05	107.4	0.52	0	15000
1	Elitism	0.05	24.8	0.05	98.9	0.51	0	15000
1	Roulette	0.05	24.8	0.05	155.0	0.60	0	15000
1	Tournament	0.1	24.8	0.05	100.6	0.51	0	15000
1	Elitism	0.1	24.8	0.05	103.6	0.52	0	15000
1	Roulette	0.1	24.8	0.05	133.5	0.59	0	15000
5	Tournament	0	360	1	178.1	0.50	0	15000
5	Elitism	0	360	1	185.9	0.53	0	15000
5	Roulette	0	360	1	200.3	0.55	0	15000
5	Tournament	0.05	360	1	185.6	0.51	0	15000
5	Elitism	0.05	360	1	188.3	0.54	0	15000
5	Roulette	0.05	360	1	265.4	0.70	0	15000
5	Tournament	0.1	360	1	180.9	0.53	0	15000
5	Elitism	0.1	360	1	183.6	0.50	0	15000
5	Roulette	0.1	360	1	228.1	0.60	0	15000

Los Gráfico 15 y Gráfico 16 muestran un patrón similar entre el costo relativo de la información y el beneficio y densidad promedio respectivamente. Como se observa en la Tabla 18 sólo un caso ha acertado para el caso de red vacía.

Gráfico 15. Costo relativo de la información y beneficio promedio (20 nodos)

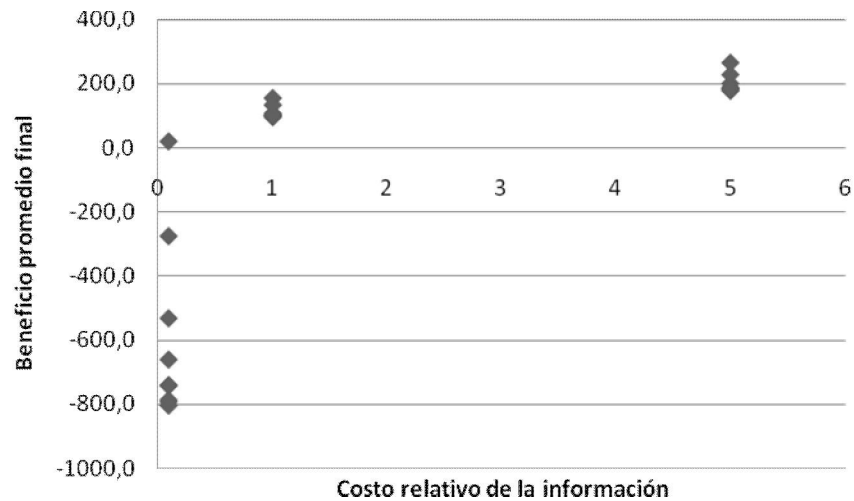
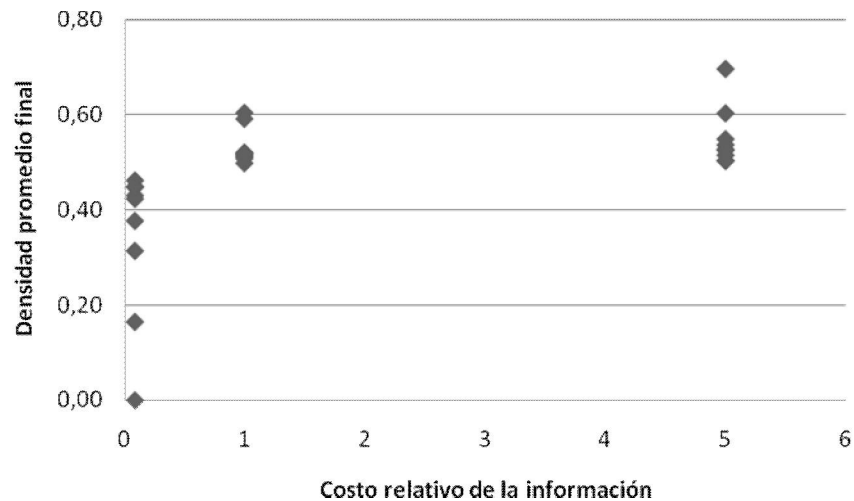


Gráfico 16. Costo relativo de la información y densidad promedio (20 nodos)



4.2.1.4 Efecto del tamaño de población

A continuación se analiza el efecto del tamaño de población (número de nodos en una red sobre las topologías encontradas. La Tabla 19 muestra algunas variables relevantes y sus valores promedio, categorizados por el tipo de flujo de información (monodireccional o bidireccional) y por el tamaño de la red (5, 10 y 20 nodos).

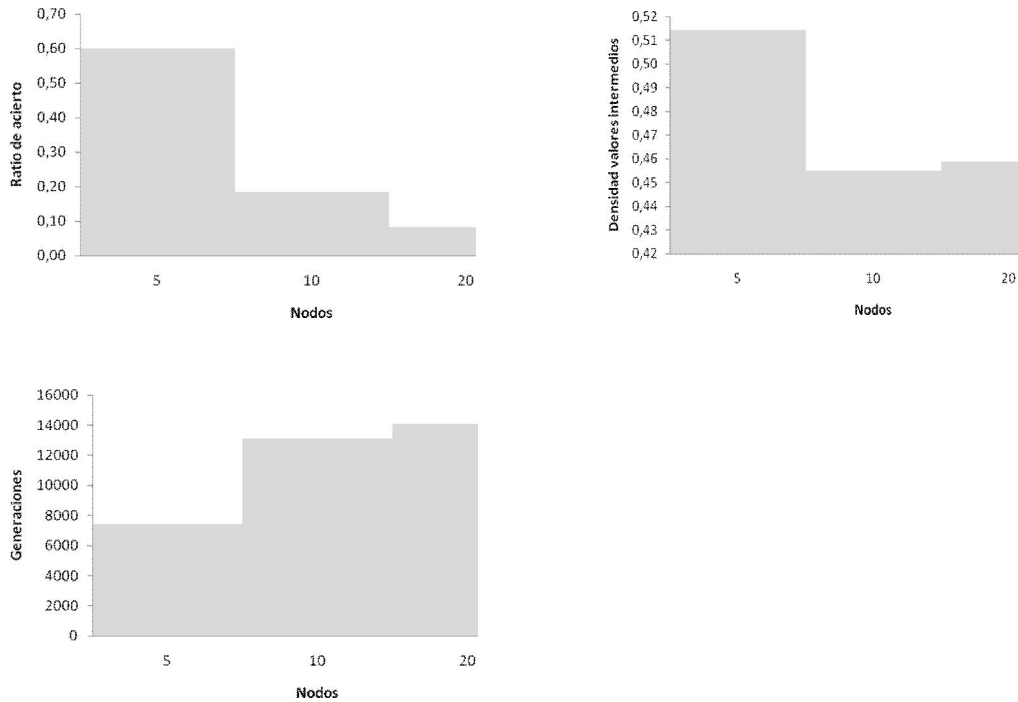
Tabla 19. Tamaño de red y su efecto en ratio de acierto, valores medios y generaciones promedio

	Ratio acierto	Densidad encontrada en valores medios	Generaciones
Monodireccional			
5 nodos	0.60	0.51	7428
10 nodos	0.19	0.45	13130
20 nodos	0.08	0.46	14055
Bidireccional			
5 nodos	0.62	0.90	1231
10 nodos	0.37	0.81	7790
20 nodos	0.04	0.53	14752

La misma información se representa a continuación en la Tabla 20 que muestra barras asociadas a los valores de las redes de comunicación monodireccional. Obsérvese en el gráfico superior derecho que el ratio de acierto disminuye con el tamaño de red para un número fijo de generaciones. Lo mismo ocurre con la densidad asociada a los valores intermedios de información (gráfico superior izquierdo). Dichos valores debieran hacer surgir una red de topología de estrella (valor de densidad aproximado a $1/n$) mostrando sin embargo que para redes pequeñas ese valor tiende a mostrar la red completa como equilibrio estable. Los valores decrecientes con el número de nodos en la red en esta variable deben asociarse a la limitación en el número de generaciones que imposibilitó llegar a un valor intermedio como en el caso de la red de 5 nodos.

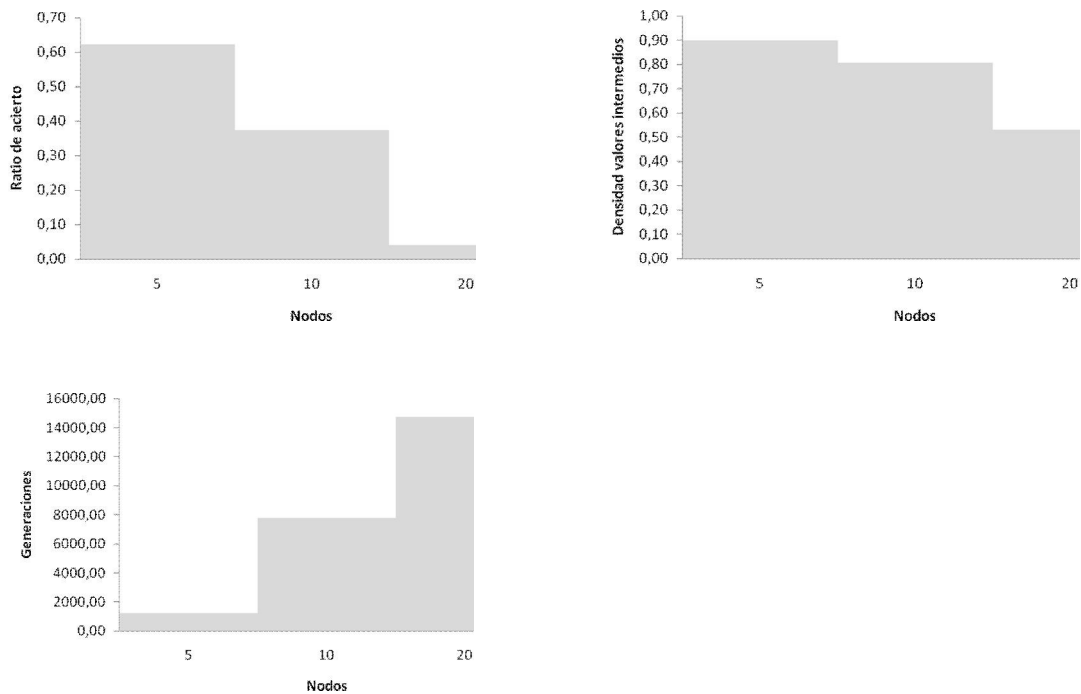
Esto último se aprecia mejor en el gráfico inferior derecho, donde se parecía claramente como son requeridos ingentes cantidades de generaciones a medida que aumenta el tamaño de la red.

Tabla 20. Ratio de acierto, Densidad de valores intermedios y generaciones promedio para comunicación monodireccional



Observaciones análogas al modelo monodireccional pueden realizarse para el caso bidireccional (Tabla 21). Obsérvese que en el gráfico superior derecho como el ratio de acierto disminuye a medida que se incrementa el tamaño de red de manera más marcada aún que en caso monodireccional. Del mismo modo, el gráfico superior izquierdo muestra una mayor variabilidad entre la densidad de la red para valores intermedios, aunque en todos los casos los valores son más altos de los esperables. Finalmente, el gráfico inferior derecho muestra un fuerte crecimiento en la necesidad de generaciones para en promedio mostrar un resultado estable.

Tabla 21. Ratio de acierto, densidad de valores intermedios y generaciones promedio para comunicación bidireccional



4.2.1.5 Efectos de la selección y mutación

Vamos a observar el efecto del tipo de selección y de la tasa de mutación sobre la efectividad del AG en detectar la topología teórica. En la Tabla 22 se aprecian como el tipo de selección ha afectado en grado de acierto del AG y la velocidad de convergencia. Los valores publicados son el promedio de todas las configuraciones analizadas, por lo que tienen un sesgo a ser bajos dados los resultados de aciertos publicados en el caso de las redes más grandes. En el modelo de comunicación monodireccional la forma de selección elitista emerge levemente superior a torneo en términos de grado de acierto y de velocidad de convergencia. Sin embargo, para comunicación bidireccional esto no se mantiene y es el método de selección por ruleta el levemente más efectivo unido a una velocidad de convergencia levemente mejor que sus contrapartes de torneo y elitismo.

Tabla 22. Procedimiento de selección y su efecto en ratio de acierto y generaciones promedio

<i>Monodireccional</i>	Ratio acierto	Generaciones
Tournament	0.51	8863.9
Elitism	0.52	8788.4
Roulette	0.28	11844.1
<i>Bidireccional</i>		
Tournament	0.53	8374.2
Elitism	0.47	7777.4
Roulette	0.56	7468.0

Por otro lado, la Tabla 23 muestra el efecto de la tasa de mutación en las mismas variables que la tabla precedente. Se observa que la tasa de mutación nula es en promedio la más efectiva, seguida de la tasa de mutación de 0.05, destacándose más aún en el caso de las generaciones necesarias para converger.

Tabla 23. Tasa de mutación y su efecto en ratio de acierto y generaciones promedio

<i>Monodireccional</i>	Ratio acierto	Generaciones
0	0.50	8,891.4
0.05	0.49	9,803.1
0.1	0.32	10,801.9
<i>Bidireccional</i>		
0	0.58	8,110.0
0.05	0.54	7,388.5
0.1	0.44	8,121.2

4.2.1.6 Conclusiones parciales

El problema combinatorial enfrentado hace que sólo cuando el espacio de soluciones es tratable (red de 5 nodos) los resultados obtenidos se aproximen más a los teóricamente esperados. A medida que crece el tamaño de población para un número dado de generaciones se comienza a perder capacidad de acierto en identificar la topología teórica

en forma completa aún cuando la tendencia de evolución tenga la pendiente con el signo esperado.

Nuevamente, los resultados muestran que el AG detecta o tiende a detectar, dadas las condiciones para ello según el modelo teórico respectivo, las topologías extremas. El AG carece de poder de discernimiento de las topologías intermedias (red de estrella centralmente sostenida) problema ya observado por la literatura.

El procedimiento de selección más efectivo para grafos dirigidos son las variantes de torneo. Para el caso de los grafos no dirigidos resulta el procedimiento de selección por ruleta el que se demuestra más efectivo en promedio.

Finalmente, la tasa de mutación revela un rol más efectivo cuando los valores de mutación son nulos o bajos.

4.2.2 Modelo de Larrosa y Tohmé

Para el modelo de comunicación monodireccional indirecta con costos de conexión indirectos (Larrosa y Tohmé, 2003b) sin decaimiento se plantea las siguientes hipótesis:

- IX. *Dada la estructura de pagos (2.5), una red de comunicación monodireccional eficiente es vacía o es circular. Específicamente si costo comparado con el pago c es menor que 1 para cada uno de los miembros ($c < 1$), entonces la red circular (ciclo) es la única red eficiente (H7). Finalmente, si el costo es mayor que 1 ($c > 1$) entonces la red vacía es la única red eficiente (H8).*

Finalmente, para el modelo de comunicación bidireccional indirecta con costos acumulativos de conexión (Larrosa y Tohmé, 2005) sin decaimiento nos plantea las siguientes hipótesis:

- X. *Dada la estructura de pagos (2.5), una red de comunicación bidireccional eficiente es vacía o es una red de línea. Específicamente si costo c es positivo pero menor que 1 ($0 < c < 1$) entonces la red de línea es red eficiente (H9). Si el costo comparado con el pago es mayor 1 ($c > 1$) la red vacía es la única red eficiente (H10).*

Tabla 24. Parámetros del AG

Número de agentes	5	10	20
Generaciones por corrida	15000		
Corridas por caso	30		
Tamaño de la población	200		
Flujo de información	Monodireccional (grafo dirigido)	Bidireccional (grafo no dirigido)	
Función de desempeño (fitness)	Función de beneficios con costo indirecto e información directa con información monodireccional	Función de beneficios con costo indirecto e información directa con información bidireccional	
Tipo de selección	Tournament	Elitism	Roulette
Estilo de cruzamiento	Cruzamiento de 2 puntos		
Tasa de cruzamiento	1		
Tasa de mutación	0	0.05	0.1
Valor relativo de la información (valor info/costo)	0.09		5

4.2.2.1 Red de 5 nodos

Tabla 25. Comunicación monodireccional, 5 nodos

ratio-info-edge	Selección	Tasa de mutación	Beneficio equilibrio	Densidad equilibrio	Densidad encontrada	Beneficio encontrado	Ratio acierto	Generaciones
0.09	Tournament	0	5	0	0.0	5.0	0.1	94
0.09	Elitism	0	5	0	0.0	5.0	0	722
0.09	Roulette	0	5	0	0.0	5.0	1	3988
0.09	Tournament	0.05	5	0	0.0	5.0	1	106
0.09	Elitism	0.05	5	0	0.0	5.0	0	387
0.09	Roulette	0.05	5	0	0.0	5.0	1	928
0.09	Tournament	0.1	5	0	0.0	5.0	1	122
0.09	Elitism	0.1	5	0	0.0	5.0	0	429
0.09	Roulette	0.1	5	0	0.0	5.0	0	2055
5	Tournament	0	110.2	1	0.7	78.3	0.88	12109
5	Elitism	0	110.2	1	1.0	110.2	1	5162
5	Roulette	0	110.2	1	0.6	71.6	1	12012
5	Tournament	0.05	110.2	1	1.0	110.2	1	424
5	Elitism	0.05	110.2	1	1.0	110.2	1	1401
5	Roulette	0.05	110.2	1	1.0	110.2	1	6766
5	Tournament	0.1	110.2	1	1.0	110.2	0.84	442
5	Elitism	0.1	110.2	1	1.0	110.2	1	1434
5	Roulette	0.1	110.2	1	1.0	110.2	1	4156

Los Gráfico 17 y Gráfico 18 muestra el grado de precisión del AG para encontrar los resultados extremos de red vacía y completa con sólo dos casos alejándose de estos resultados generales.

Gráfico 17. Beneficio promedio y costo relativo de la información (5 nodos)

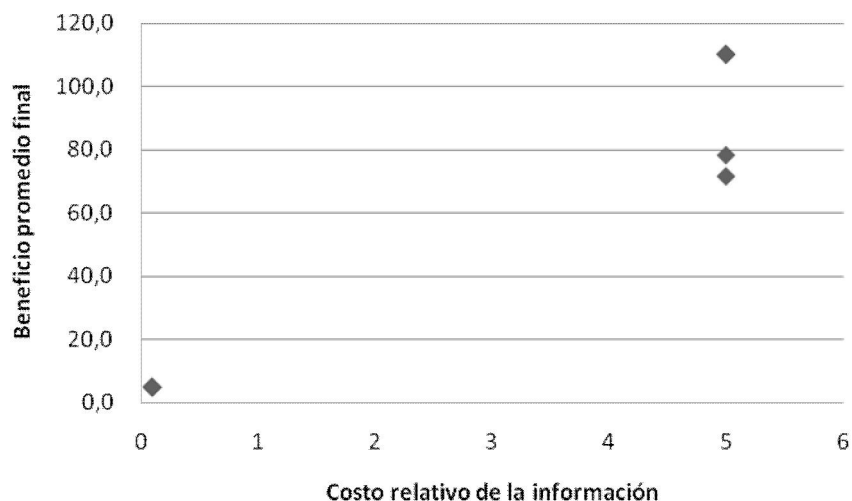
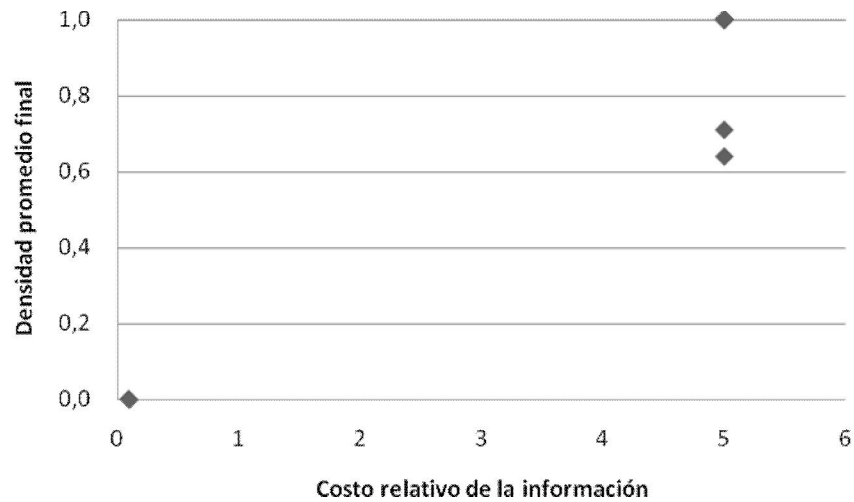
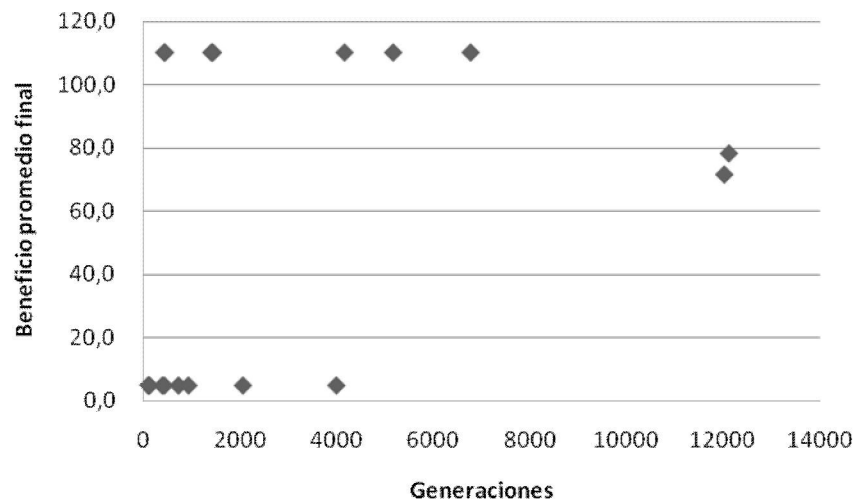


Gráfico 18. Densidad promedio y costo relativo de la información (5 nodos)



Como se observa en el Gráfico 19, no ha habido escasez de cantidad de generaciones necesarias para evolucionar hacia el resulta teórico esperado. La mayor parte de los puntos se apiñan tanto en un topología extrema (red vacía) como en la otra (red completa).

Gráfico 19. Beneficio promedio y generaciones (5 nodos)



Ahora pasemos a observar el modelo de **comunicación bidireccional** con costos indirectos de conexión y acceso a información indirecta (Tabla 26).

Tabla 26. Comunicación bidireccional, 5 nodos

ratio-info-edge	Selección	Tasa de mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio Acierto	Generaciones
0.09	Tournament	0	5	0	-163.4	0.3	0.8	15000.0
0.09	Elitism	0	5	0	-53.2	0.2	0.9	299.6
0.09	Roulette	0	5	0	5.0	0.0	1.0	390.8
0.09	Tournament	0.05	5	0	5.0	0.0	1.0	98.2
0.09	Elitism	0.05	5	0	-53.2	0.2	0.9	157.2
0.09	Roulette	0.05	5	0	5.0	0.0	1.0	285.2
0.09	Tournament	0.1	5	0	5.0	0.0	1.0	94.4
0.09	Elitism	0.1	5	0	-53.2	0.2	0.9	91.0
0.09	Roulette	0.1	5	0	-53.2	0.2	0.9	186.4
5	Tournament	0	117.6	1	105.1	0.9	0.8	6003.8
5	Elitism	0	117.6	1	117.6	1.0	1.0	389.4
5	Roulette	0	117.6	1	117.6	1.0	1.0	359.2
5	Tournament	0.05	117.6	1	117.6	1.0	1.0	255.2
5	Elitism	0.05	117.6	1	117.6	1.0	1.0	248.6
5	Roulette	0.05	117.6	1	117.6	1.0	1.0	179.0
5	Tournament	0.1	117.6	1	95.1	0.8	0.8	200.2
5	Elitism	0.1	117.6	1	117.6	1.0	1.0	304.0
5	Roulette	0.1	117.6	1	117.6	1.0	1.0	240.2

Gráfico 20. Beneficio promedio y valor relativo de la información (5 nodos)

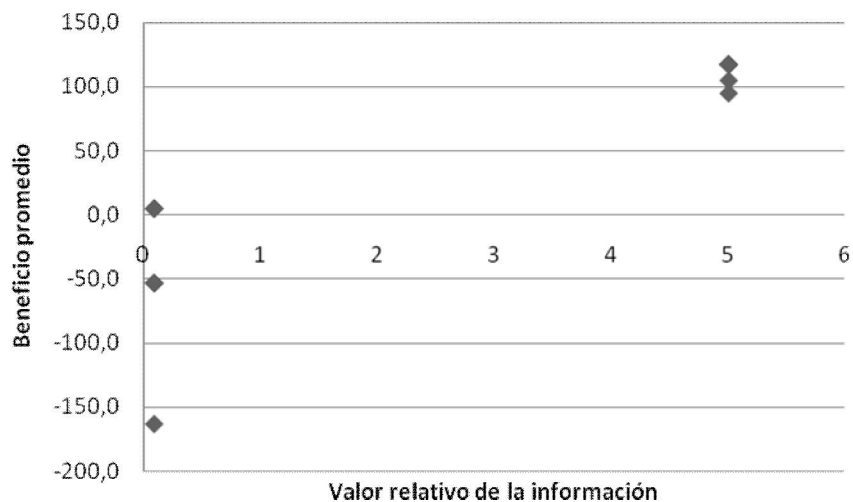
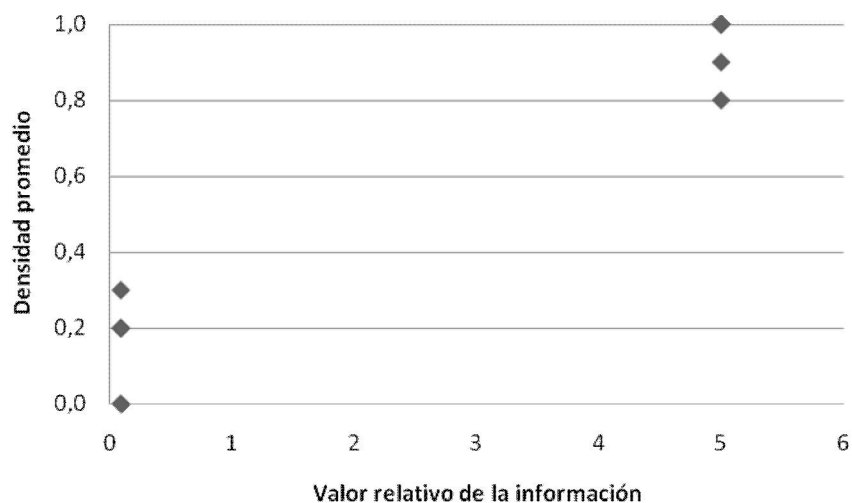


Gráfico 21. Densidad promedio y valor relativo de la información (5 nodos)



4.2.2.2 Red de 10 nodos

La Tabla 27 muestra los resultados de las simulaciones para estructuras de 10 nodos.

Tabla 27. Comunicación monodireccional, 10 nodos

ratio-info-edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.09	Tournament	0	10	0	10	0.00	1	854
0.09	Elitism	0	10	0	10	0.00	1	3522.8
0.09	Roulette	0	10	0	-1544	0.19	0	15000
0.09	Tournament	0.05	10	0	-195	0.04	0	12328.2
0.09	Elitism	0.05	10	0	10	0.00	1	10996.6
0.09	Roulette	0.05	10	0	-1584	0.20	0	15000
0.09	Tournament	0.1	10	0	-1371	0.16	0	15000
0.09	Elitism	0.1	10	0	-1280	0.17	0	15000
0.09	Roulette	0.1	10	0	-2156	0.26	0	15000
5	Tournament	0	890.2	1	459	0.55	0	15000
5	Elitism	0	890.2	1	794.52	0.87	0	15000
5	Roulette	0	890.2	1	375.44	0.53	0	15000
5	Tournament	0.05	890.2	1	842.04	0.95	0	15000
5	Elitism	0.05	890.2	1	890.2	1.00	1	7944
5	Roulette	0.05	890.2	1	556.52	0.69	0	15000
5	Tournament	0.1	890.2	1	644	0.80	0	15000
5	Elitism	0.1	890.2	1	683	0.79	0	15000
5	Roulette	0.1	890.2	1	459	0.62	0	15000

Los Gráfico 22 y Gráfico 23 muestran la relación entre el beneficio promedio y el costo relativo de la información. A medida que el costo disminuye los beneficios crecen. Obsérvese que a costos relativos altos los beneficios se hacen altamente negativos y la red más mala.

Gráfico 22. Beneficio promedio y costo relativo de la información (10 nodos)

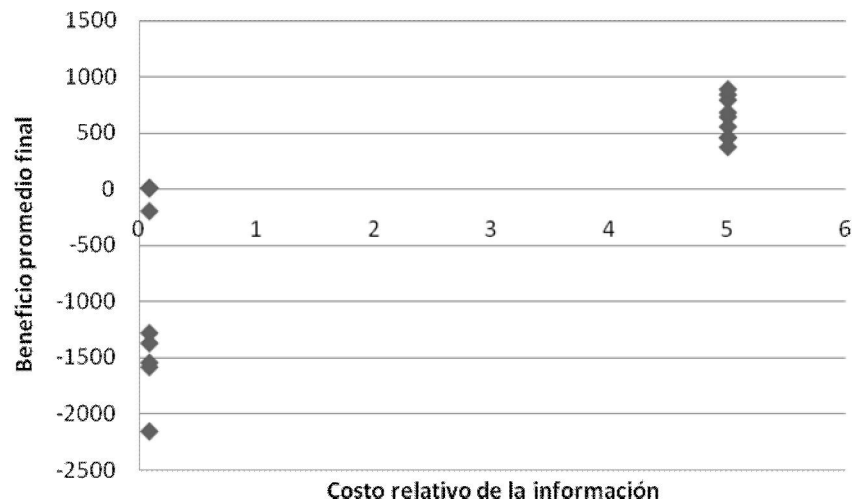
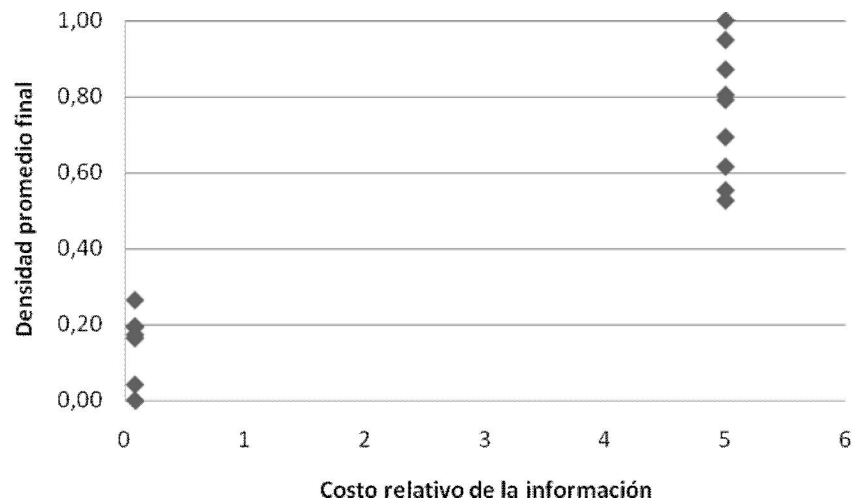
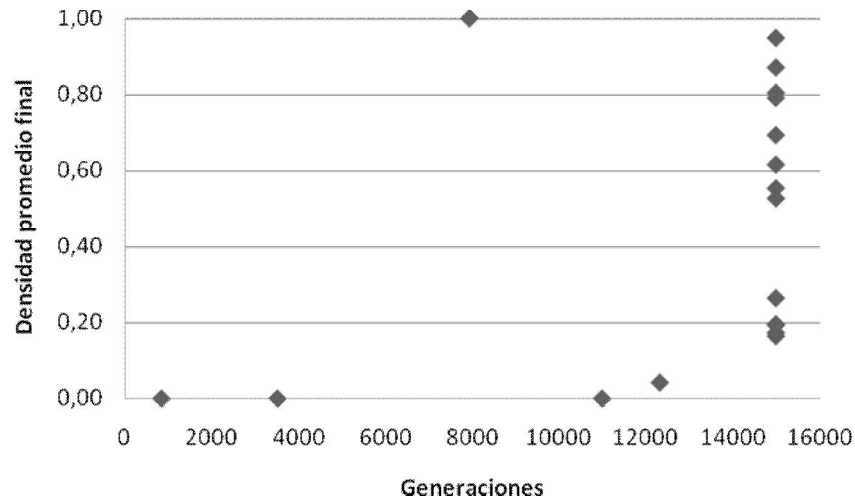


Gráfico 23. Densidad promedio y costo relativo de la información (10 nodos)



El Gráfico 24 muestra una imagen repetida. El relativamente escaso número de generaciones no permite converger el resultado observado al teórico en muchos casos.

Gráfico 24. Densidad promedio y generaciones (10 nodos)



Ahora pasemos a observar el modelo de **comunicación bidireccional** con costos indirectos de conexión y acceso a información indirecta (Tabla 28).

Tabla 28. Comunicación bidireccional, 10 nodos

ratio-info-edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.09	Tournament	0	10	0	-1503	0.41	0	15000
0.09	Elitism	0	10	0	-654	0.20	0	9078.8
0.09	Roulette	0	10	0	10	0.00	1	1694.4
0.09	Tournament	0.05	10	0	10	0.00	1	4763.6
0.09	Elitism	0.05	10	0	-594	0.16	0	9703
0.09	Roulette	0.05	10	0	10	0.00	1	3629.8
0.09	Tournament	0.1	10	0	10	0.00	1	6627.6
0.09	Elitism	0.1	10	0	-1400	0.32	0	15000
0.09	Roulette	0.1	10	0	-107	0.04	0	15000
5	Tournament	0	945.1	1	427	0.56	0	15000
5	Elitism	0	945.1	1	430.84	0.52	0	15000
5	Roulette	0	945.1	1	945.1	1.00	1	3424.2
5	Tournament	0.05	945.1	1	945.1	1.00	1	8242
5	Elitism	0.05	945.1	1	495.64	0.62	0	15000
5	Roulette	0.05	945.1	1	945.1	1.00	1	5321.8
5	Tournament	0.1	945.1	1	796	0.86	0	15000
5	Elitism	0.1	945.1	1	555	0.69	0	15000
5	Roulette	0.1	945.1	1	798	0.87	0	15000

Los Gráfico 25 y Gráfico 26 muestran la relación entre el beneficio y densidad promedio con el costo de la información. Es un resultado similar a los hallados en el caso monodireccional.

Gráfico 25. Beneficio promedio y costo relativo de la información (10 nodos)

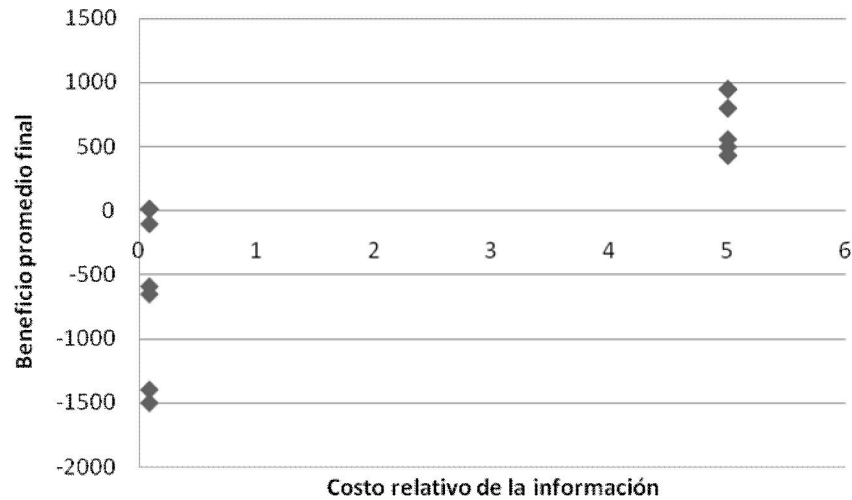
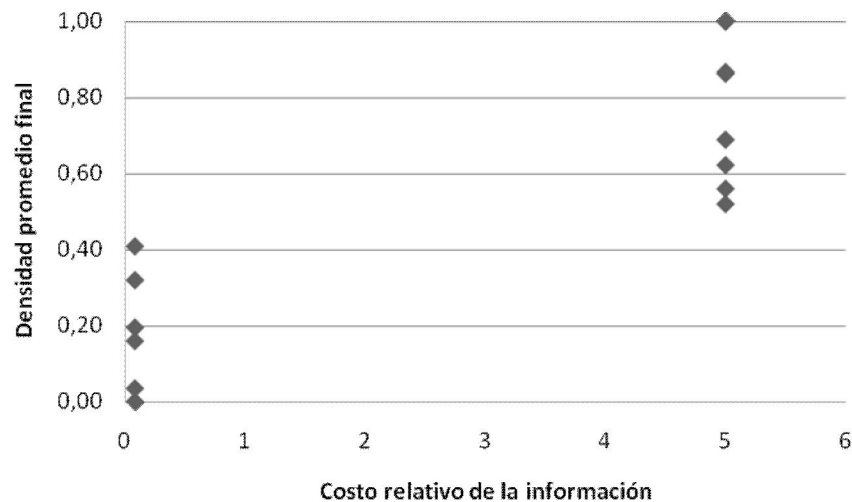
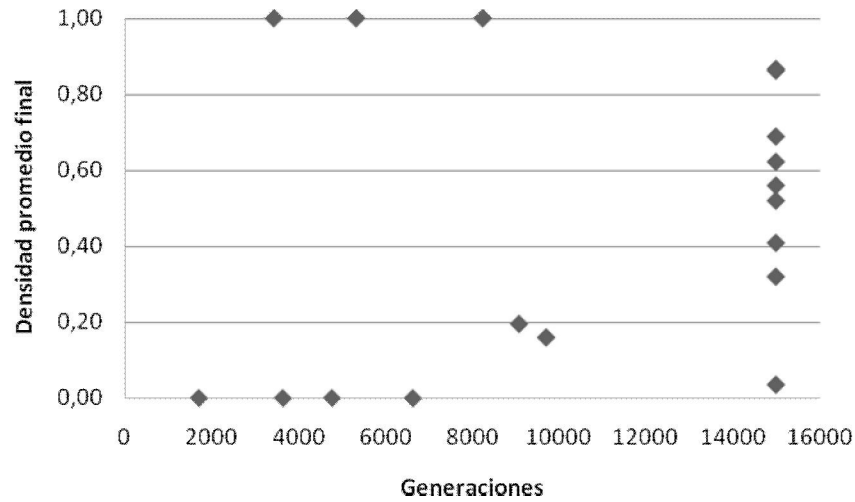


Gráfico 26. Densidad promedio y costo relativo de la información (10 nodos)



La misma referencia se puede hacer respecto al Gráfico 27. Se aprecia que el número de generaciones ha hecho que muchas simulaciones no terminen de converger.

Gráfico 27. Densidad promedio y generaciones (10 nodos)



Veamos ahora los resultados para redes de 20 nodos.

4.2.2.3 Red de 20 nodos

La Tabla 29 muestra los resultados de las simulaciones para los parámetros correspondientes.

Tabla 29. Comunicación monodireccional, 20 nodos

ratio-info -edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.09	Tournament	0	20	0	-68	0.01	0	15000
0.09	Elitism	0	20	0	-2,251	0.03	0	15000
0.09	Roulette	0	20	0	-25,263	0.32	0	15000
0.09	Tournament	0.05	20	0	-23,644	0.38	0	15000
0.09	Elitism	0.05	20	0	-24,567	0.38	0	15000
0.09	Roulette	0.05	20	0	-27,843	0.39	0	15000
0.09	Tournament	0.1	20	0	-27,212	0.42	0	15000
0.09	Elitism	0.1	20	0	-28,438	0.41	0	15000
0.09	Roulette	0.1	20	0	-30,952	0.45	0	15000
5	Tournament	0	7560	1	3,056	0.53	0	15000
5	Elitism	0	7560	1	4,484	0.66	0	15000
5	Roulette	0	7560	1	2,852	0.51	0	15000
5	Tournament	0.05	7560	1	-23,644	0.38	0	15000
5	Elitism	0.05	7560	1	-24,567	0.38	0	15000

5	Roulette	0.05	7560	1	-27,843	0.39	0	15000
5	Tournament	0.1	7560	1	4,021	0.64	0	15000
5	Elitism	0.1	7560	1	4,021	0.65	0	15000
5	Roulette	0.1	7560	1	3,279	0.56	0	15000

El Gráfico 28 y el Gráfico 29 muestran como el grado de aproximación de los resultados a la topología de equilibrio y como fallan en llegar al valor de beneficios 20 y al valor de densidad de red igual a cero. Lo mismo ocurre para valores altos. No logran obtener los beneficios ni la densidad de la red completa.

Gráfico 28. Beneficio promedio y costo relativo de la información (20 nodos)

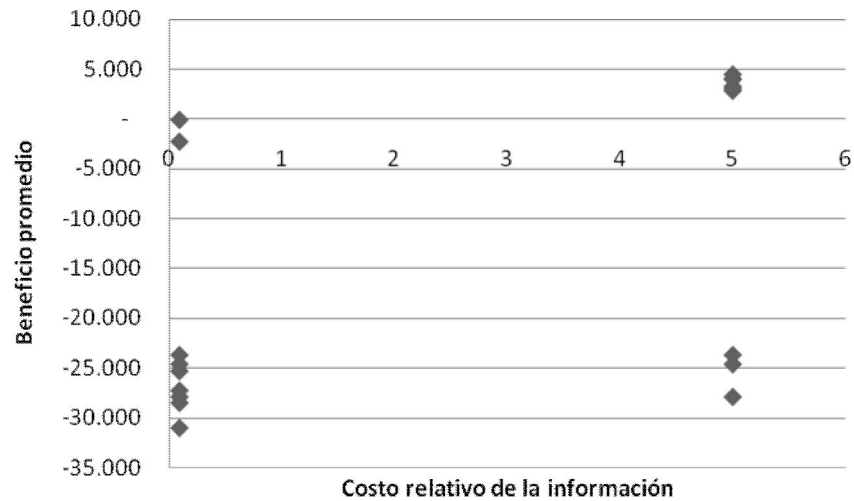
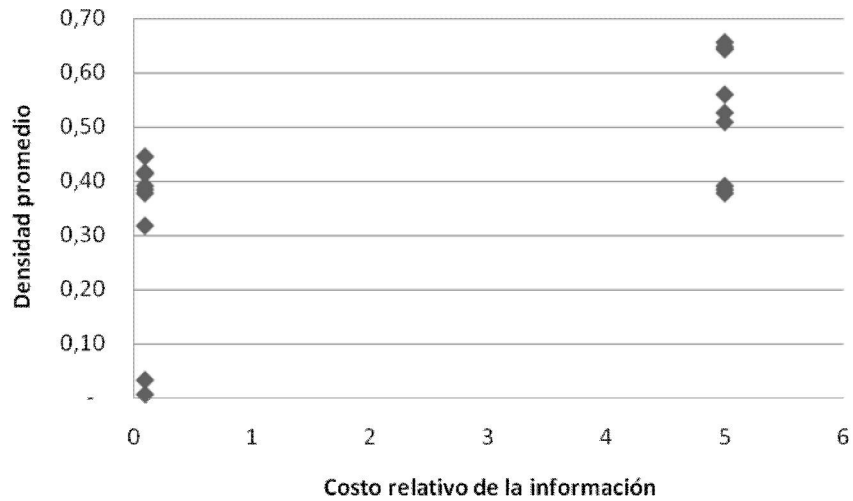
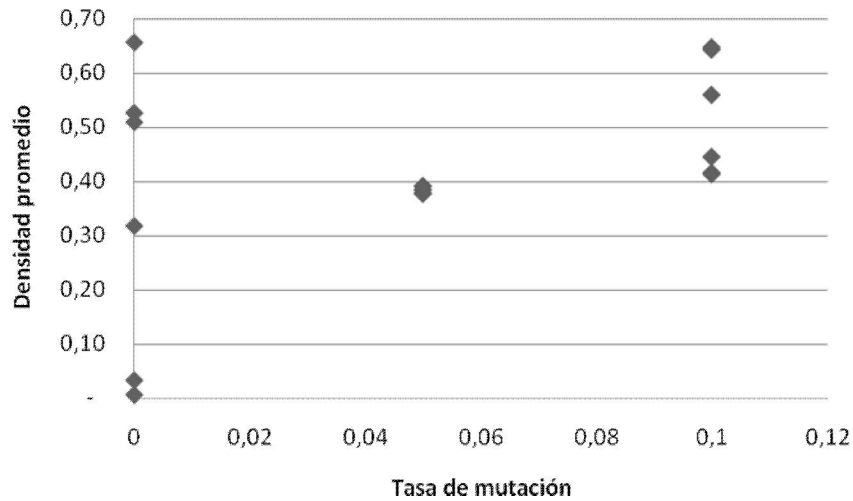


Gráfico 29. Densidad de red y costo relativo de información (20 nodos)



Finalmente el Gráfico 30 muestra como la tasa de mutación alta está asociada a redes más densas mientras que, extrañamente el valor intermedio entre las tasas de mutación utilizadas concentran valores intermedios en la densidad de red.

Gráfico 30. Densidad de red y tasa de mutación (20 nodos)



Ahora pasemos a analizar el caso de flujo de **información bidireccional**. La Tabla 30 muestra los resultados de las simulaciones para 20 nodos.

Tabla 30. Comunicación bidireccional, 20 nodos

ratio-info -edge	Selección	Mutación	Beneficio equilibrio	Densidad equilibrio	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
0.09	Tournament	0	20	0	-13,417	0.47	0	15000
0.09	Elitism	0	20	0	-10,876	0.34	0	15000
0.09	Roulette	0	20	0	-1,026	0.04	0.1	15000
0.09	Tournament	0.05	20	0	-11,142	0.36	0	15000
0.09	Elitism	0.05	20	0	-14,238	0.41	0	15000
0.09	Roulette	0.05	20	0	-14,250	0.43	0	15000
0.09	Tournament	0.1	20	0	-13,423	0.40	0	15000
0.09	Elitism	0.1	20	0	-15,160	0.47	0	15000
0.09	Roulette	0.1	20	0	-14,025	0.44	0	15000
5	Tournament	0	7320	1	3,208	0.53	0	15000
5	Elitism	0	7320	1	3,195	0.52	0	15000
5	Roulette	0	7320	1	3,409	0.55	0	15000
5	Tournament	0.05	7320	1	3,299	0.53	0	15000
5	Elitism	0.05	7320	1	3,251	0.53	0	15000
5	Roulette	0.05	7320	1	5,382	0.76	0	15000
5	Tournament	0.1	7320	1	3,073	0.51	0	15000
5	Elitism	0.1	7320	1	3,242	0.53	0	15000
5	Roulette	0.1	7320	1	4,274	0.65	0	15000

El Gráfico 31 muestra cómo se agrupan los valores altos de información con niveles más altos de beneficios. Algo similar se observa en el Gráfico 32 con respecto a la densidad promedio.

Gráfico 31. Beneficio promedio y costo relativo de la información (20 nodos)

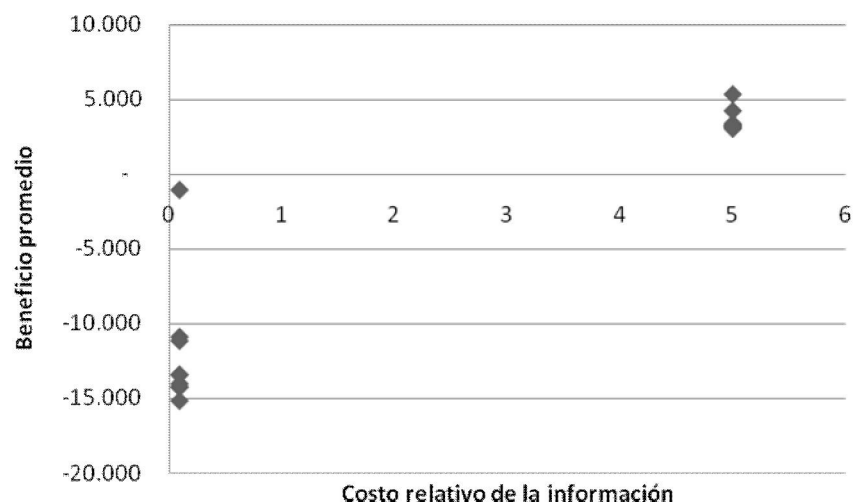
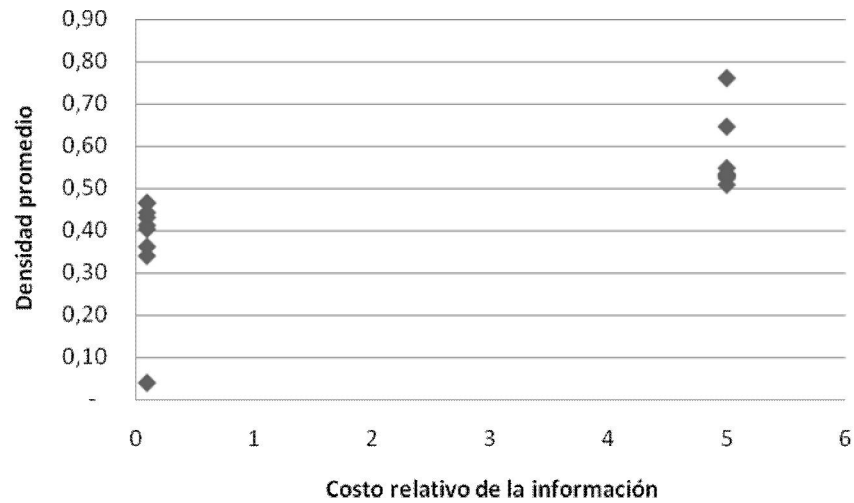
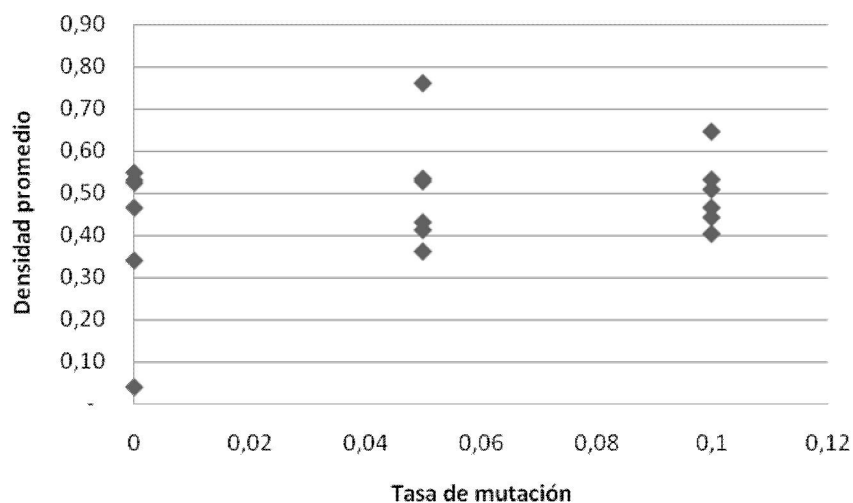


Gráfico 32. Densidad promedio y costo relativo de la información (20 nodos)



El Gráfico 33 muestra una relación de las densidades de red más concentradas a medida que aumenta la tasa de mutación. Grandes niveles de experimentación no dejan definir claramente una topología extrema. Para valores nulos algunos corridas se aproximan a la red vacía, aunque en general independientemente de la tasa de mutación la mayor parte de los resultados arroja la presencia de una red de densidad intermedia.

Gráfico 33. Densidad promedio y tasa de mutación (20 nodos)



4.2.2.4 Efecto del tamaño de población

A continuación se analiza el efecto del tamaño de población (número de nodos en una red sobre las topologías encontradas. La Tabla 31 muestra algunas variables relevantes y sus valores promedio, categorizados por el tipo de flujo de información (monodireccional o bidireccional) y por el tamaño de la red (5, 10 y 20 nodos) para el modelo de costos acumulativos e información indirecta.

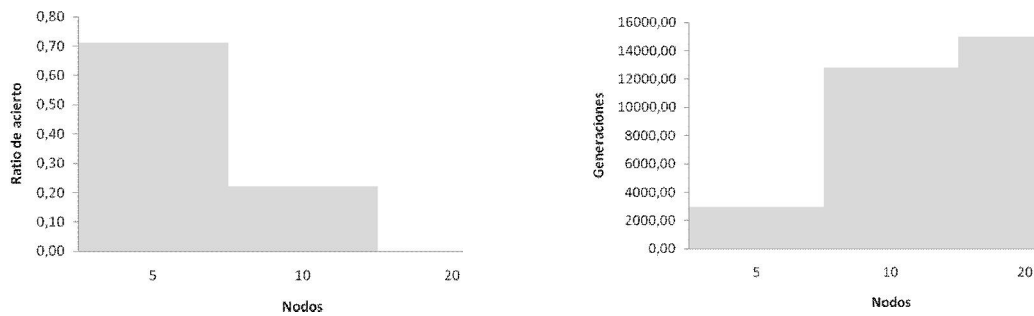
Tabla 31. Tamaño de red y su efecto en ratio de acierto y generaciones promedio

<i>Monodireccional</i>	Ratio acierto	Generaciones
5 nodos	0.71	2929.83
10 nodos	0.22	12813.64
20 nodos	0.00	15000.00
<i>Bidireccional</i>		
5 nodos	0.94	1376.80
10 nodos	0.39	10415.84
20 nodos	0.01	15000.00

La misma información se representa a continuación en la Tabla 32 que muestra gráficos en barras asociadas a los valores de las redes de comunicación monodireccional. Obsérvese en

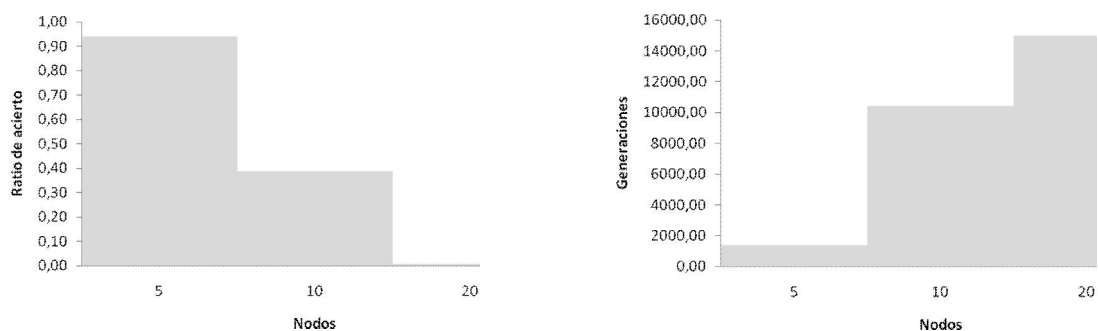
el gráfico superior derecho que el ratio de acierto disminuye con el tamaño de red para un número fijo de generaciones. Algo inverso se aprecia mejor en el gráfico superior izquierdo, donde se aprecia claramente como son requeridos ingentes cantidades de generaciones a medida que aumenta el tamaño de la red.

Tabla 32. Ratio de acierto y generaciones promedio para comunicación monodireccional



Observaciones análogas al modelo monodireccional pueden realizarse para el caso bidireccional (Tabla 33). Obsérvese que en el gráfico superior derecho como el ratio de acierto disminuye a medida que se incrementa el tamaño de red de manera más marcada aún que en caso monodireccional. Del mismo modo, el gráfico superior izquierdo muestra un fuerte crecimiento en la necesidad de generaciones en promedio para mostrar un resultado estable.

Tabla 33. Ratio de acierto, Densidad de valores intermedios y generaciones promedio para comunicación bidireccional



Veamos a continuación los efectos del proceso de selección y mutación en la eficiencia del AG.

4.2.2.5 Efectos de la selección y mutación

Finalmente analicemos el efecto del tipo de selección y de la tasa de mutación sobre la efectividad del AG en detectar la topología teórica. En la Tabla 34 se aprecian como el tipo de selección ha afectado en grado de asertividad del AG y la velocidad de convergencia. Los valores publicados son el promedio de todas las configuraciones analizadas, por lo que tienen un sesgo a ser bajos dados los resultados de aciertos publicados en el caso de las redes más grandes. En el modelo de comunicación monodireccional la forma de selección elitista emerge levemente superior a torneo en términos de grado de acierto y algo más significativamente en términos de velocidad de convergencia. Sin embargo, para comunicación bidireccional esto no se mantiene y es el método de selección por ruleta el más efectivo unido a una velocidad de convergencia muy superior a sus contrapartes de torneo y elitismo.

Tabla 34. Procedimiento de selección y su efecto en ratio de acierto y generaciones promedio para comunicación bidireccional

<i>Monodireccional</i>	Ratio acierto	Generaciones
Tournament	0.32	9804.4
Elitism	0.33	9277.7
Roulette	0.28	11661.4
<i>Bidireccional</i>		
Tournament	0.46	9793.6
Elitism	0.32	9459.5
Roulette	0.56	7539.5

Por otro lado, la Tabla 35 muestra el efecto de la tasa de mutación en las mismas variables que la tabla precedente. Se observa que la tasa de mutación de 0.05 es en promedio la más efectiva, seguida de la tasa de mutación nula.

Tabla 35. Tasa de mutación y su efecto en ratio de acierto y generaciones promedio para comunicación bidireccional

<i>Monodireccional</i>	Ratio acierto	Generaciones
0	0.33	10470.2
0.05	0.39	9793.4
0.1	0.21	10479.9
<i>Bidireccional</i>		
0	0.42	9535.6
0.05	0.55	7660.2
0.1	0.08	14302.3

4.2.2.6 Conclusiones parciales

Nuevamente se presenta el problema combinatorial: cuando la red es más pequeña los resultados se aproximan a los valores teóricos más rápidamente. A medida que crece el tamaño de población para un número dado de generaciones se comienza a perder capacidad de acierto en identificar la topología teórica en forma completa aún cuando la tendencia de evolución tenga la pendiente con el signo esperado.

Entre los procedimientos de selección para grafos dirigidos los tres mantienen niveles de efectividad similares con una leve tendencia superior para el caso de las variantes de torneo.

Para el caso de los grafos no dirigidos resulta el procedimiento de selección por ruleta el que se demuestra más efectivo en promedio, seguido del torneo.

Finalmente, la tasa de mutación revela un rol más efectivo cuando los valores de mutación son bajos aunque no nulos.

4.2.3 Modelo de Goyal y Joshi

Ahora iniciamos los modelos de formación de redes de colaboración de Goyal y Joshi (2003).

En estos modelos la red disminuye los costos de producción. La función que se optimiza es una función de producción tradicional derivada de un modelo de Cournot. En este caso, Goyal y Joshi trabajan particularmente con un grafo no dirigido, haciendo que las conexiones requieran de mutuo consentimiento de las partes. Las hipótesis serían:

HIPÓTESIS

- XI. *Existiendo comunicación bidireccional, una red g es una red eficiente si $n_i(g) \in \{0, n-1\}$ para todo $i \in N$. Más aún,*
- si el costo de realizar un enlace, φ es bajo entonces g_c es la única red eficiente (H11);*
 - si el costo de realizar un enlace, φ es alto entonces g_e es la única red eficiente (H12) (Goyal & Joshi, 2003: 61)*

Prueba: Ver Goyal y Joshi (2003: 606).

Para este modelo se parametrizaron dos escenarios, uno con un φ alto y otro con un φ bajo, esperando corroborar si el código detectaba los cambios en las topología de equilibrio (Tabla 36).

Tabla 36. Parámetros del AG

Número de agentes	15	30
Generaciones por corrida	15000	
Corridas por caso	50	
Tamaño de la población	200	
Flujo de información	Bidireccional (grafo no dirigido)	
Función de desempeño (<i>fitness</i>)	Función de beneficios con reducción de costos bilateral	
Tipo de selección	Tournament	Elitism
Estilo de cruzamiento	Cruzamiento de 2 puntos	
Tasa de cruzamiento	1	
Tasa de mutación	0	0.05
α	60	17
γ	.035	0.21
γ_0	4	50
φ	0.16	20

4.2.3.1 Red de 15 nodos

La Tabla 37 muestra los resultados para un φ bajo.

Tabla 37. Resultados de las simulaciones para un φ bajo (15 nodos)

Selección	Mutación	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
Tournament	0	400072.8	0.532	0.00	15000
Elitism	0	400145.0	0.697	0.22	14201
Roulette	0	400277.4	1.00	1.00	5045
Tournament	0.05	400173.7	0.763	0.00	15000
Elitism	0.05	400099.7	0.594	0.00	15000
Roulette	0.05	400230.3	0.892	0.00	15000
Tournament	0.1	400106.0	0.608	0.00	15000
Elitism	0.1	400081.6	0.552	0.00	15000
Roulette	0.1	400174.8	0.765	0.00	15000

El Gráfico 34 muestra como los niveles de beneficio promedio tienden a agruparse a medida que aumenta la tasa de mutación. Algo similar ocurre con la tasa de mutación y la densidad promedio final según se aprecia en el Gráfico 35.

Gráfico 34. Beneficio promedio y tasa de mutación (15 nodos)

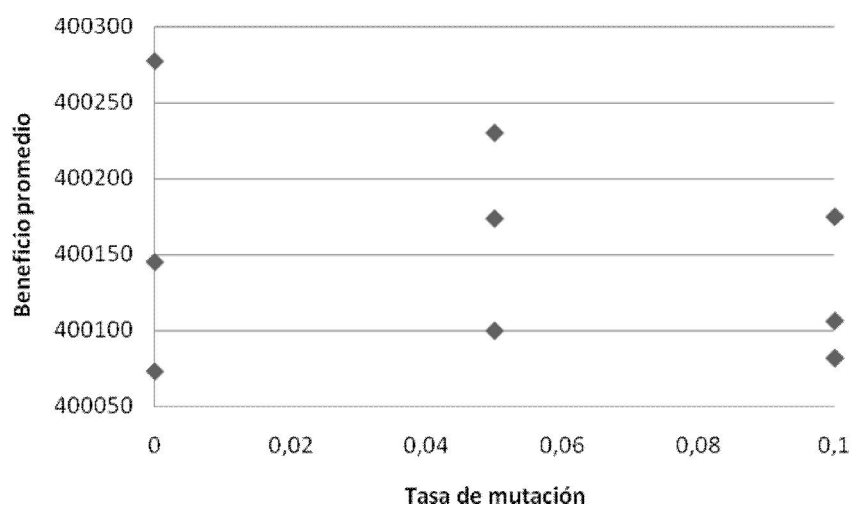
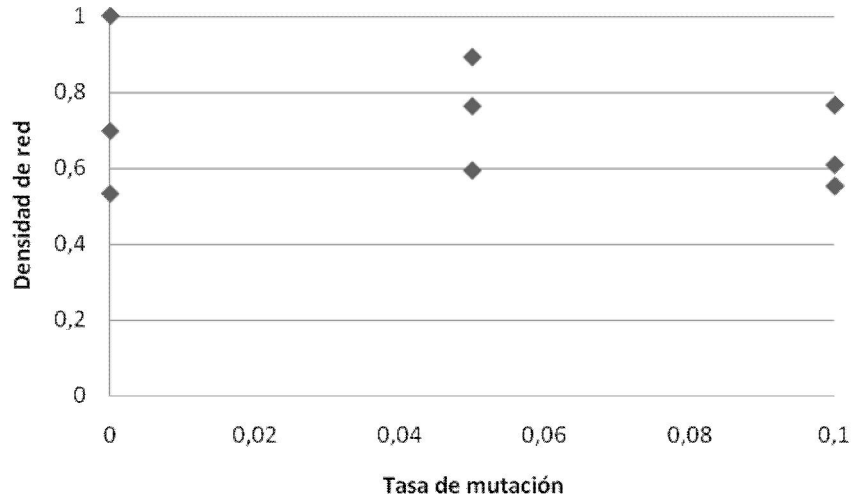
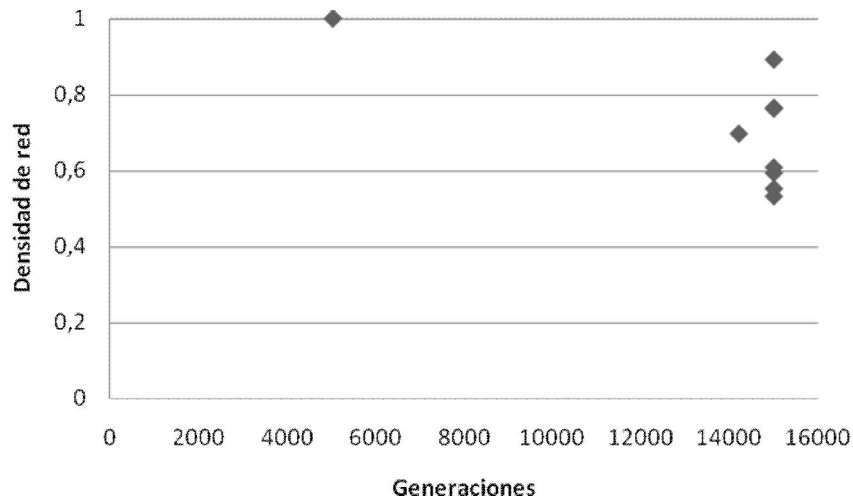


Gráfico 35. Densidad de red promedio y tasa de mutación (15 nodos)



El Gráfico 36 también muestra que gran de las corridas se agolpan en varios medios de densidad al final del tope de generaciones, revelando la necesidad de mayor campo de exploración para esta cantidad de nodos.

Gráfico 36. Densidad de red promedio y generaciones (15 nodos)



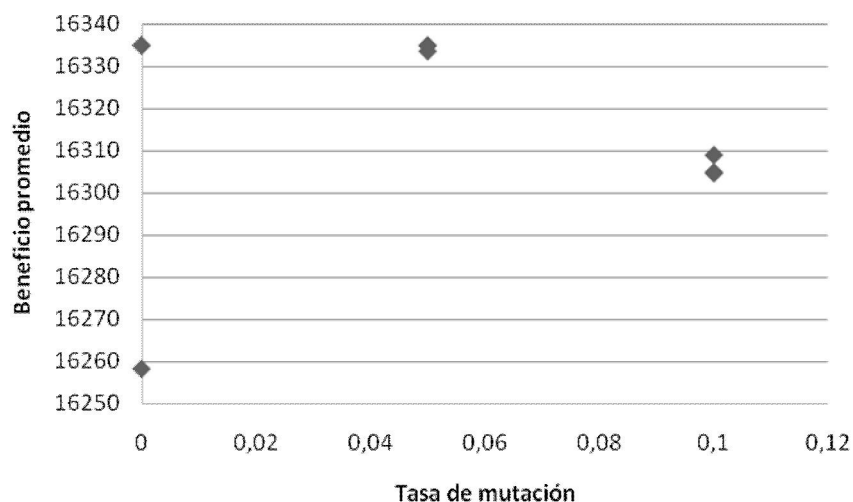
Ahora Tabla 38 muestra los resultados para el caso de un ϕ alto.

Tabla 38. Resultados de las simulaciones para un ϕ alto (15 nodos)

Selección	Mutación	Beneficio encontrado	Densidad encontrada	Ratio Acierto	Generaciones
Tournament	0	16258.2	0.438	0.0	20000
Elitism	0	16335	0	1.0	1194
Roulette	0	16335	0	1.0	1354.8
Tournament	0.05	16333.7	0.008	0.8	16705.6
Elitism	0.05	16335	0	1.0	9449.2
Roulette	0.05	16335	0	1.0	8873.8
Tournament	0.1	16309.0	0.151	0.0	20000
Elitism	0.1	16304.9	0.171	0.0	20000
Roulette	0.1	16304.7	0.175	0.0	20000

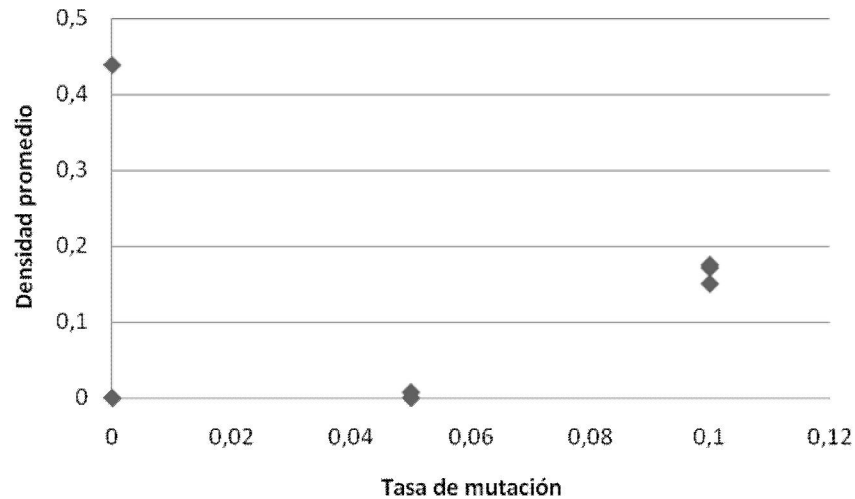
El Gráfico 37 muestra la relación entre el beneficio promedio y la tasa de mutación del ejercicio computacional. Existe una convergencia en los puntos hacia un nivel de beneficio intermedio a medida que se hace más grande la tasa de mutación.

Gráfico 37. Beneficio promedio y tasa de mutación (15 nodos)



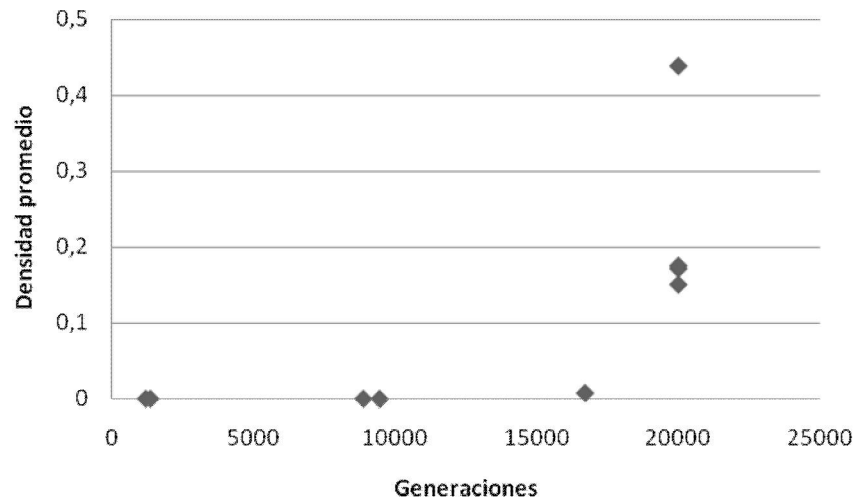
El Gráfico 38 muestra una escasa relación entre la densidad promedio y la tasa de mutación.

Gráfico 38. Densidad promedio y tasa de mutación (15 nodos)



Por otro lado, el Gráfico 39 revela una relación de escasez de generaciones para alcanzar la convergencia a los resultados del modelo teórico.

Gráfico 39. Densidad promedio y generaciones (15 nodos)



4.2.3.2 Red de 30 nodos

Finalmente la Tabla 39 muestra los resultados para la red de 30 nodos. Simplemente el número de generaciones resultó escaso para dejar una topología distinta a un red mediamente conectada. Sólo bajo una tasa de mutación de 0.05 y regla de selección por ruleta el valor se acercó a 1.

Tabla 39. Resultados de las simulaciones para un φ bajo (30 nodos)

Selección	Mutación	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
Tournament	0	1506181.57	0.51545338	0	15000
Elitism	0	1506193.71	0.51885057	0	15000
Roulette	0	1506247.89	0.54965517	0	15000
Tournament	0.05	1506181.54	0.51195402	0	15000
Elitism	0.05	1506207.44	0.52666667	0	15000
Roulette	0.05	1506559.97	0.72689655	0	15000
Tournament	0.1	1506198.14	0.52137931	0	15000
Elitism	0.1	1506213.15	0.52988506	0	15000
Roulette	0.1	1506362.05	0.61448276	0	15000

4.2.4 Modelo de Billand y Bravard

Seguimos con el último modelo de formación de redes de colaboración de Billand y Bravard (2004). En este modelo también la red disminuye los costos de producción aunque en este caso trabajan particularmente con un grafo dirigido, haciendo que las conexiones no requieran de mutuo consentimiento de las partes. Las hipótesis serían:

- XII. *Existiendo comunicación monodireccional, una red g es una red eficiente si $n_i(g) \in \{0, n-1\}$ para todo $i \in N$. Más aún,*
- si el costo de realizar un enlace, φ es bajo entonces g_c es la única red eficiente (H13);*
 - si el costo de realizar un enlace, φ es alto entonces g_e es la única red eficiente (H14) (Billand & Bravard, 2004: 600)*

La Tabla 40 expone los parámetros de los AG a desarrollar. Los valores de los parámetros expuestos en la tabla se corresponden con ejemplo 1 del trabajo de Billand y Bravard (2004).

El resultado esperable es la red vacía.

Tabla 40. Parámetros del AG

Número de agentes	10	20	
Generaciones por corrida	20000		
Corridas por caso	50		
Tamaño de la población	200		
Flujo de información	Monodireccional (grafo dirigido)		
Función de desempeño (<i>fitness</i>)	Función de beneficios con reducción de costos unilateral		
Tipo de selección	Tournament	Elitism	Roulette
Estilo de cruzamiento	Cruzamiento de 2 puntos		
Tasa de cruzamiento	1		
Tasa de mutación	0	0.05	0.1
α	15000		
γ	2000		
γ_0	100		
φ	50000		

4.2.4.1 Red de 10 nodos

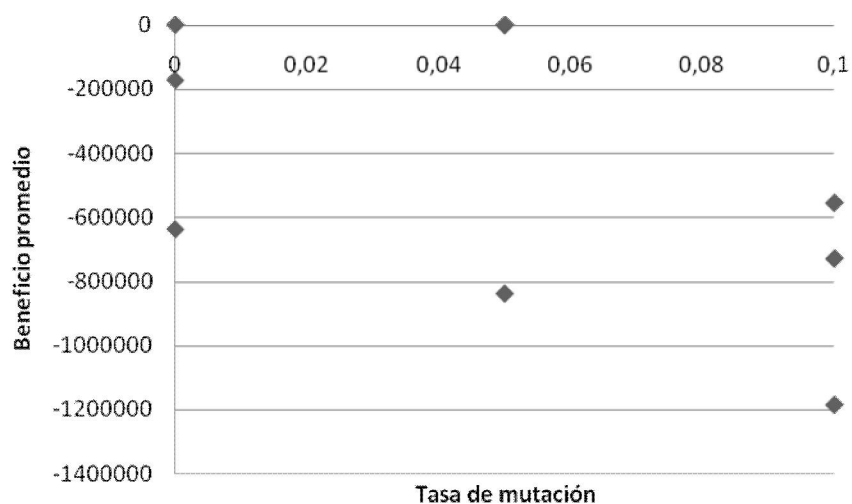
Ahora analizaremos el caso de la red de 10 nodos (Tabla 41). Se observa que el ratio de aciertos es más alto en los procesos de selección más competitivos (elitismo y torneo) y con preeminencia de una moderada tasa de mutación.

Tabla 41. Resultados de las simulaciones, 10 nodos

Selección	Mutación	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
Tournament	0	-172500	0.04	0.05	19012.8
Elitism	0	0	0	1.00	1499.5
Roulette	0	-637500	0.142	0.00	20000
Tournament	0.05	0	0	1.00	5329.2
Elitism	0.05	0	0	1.00	4296.6
Roulette	0.05	-837500	0.186	0.00	20000
Tournament	0.1	-555000	0.123	0.00	20000
Elitism	0.1	-728947.4	0.162	0.00	20000
Roulette	0.1	-1185000	0.263	0.00	20000

El Gráfico 40 muestra la relación entre el beneficio promedio y la tasa de mutación. Se observa que el número de generaciones resultó escaso para hacer converger al modelo teórico.

Gráfico 40. Beneficios promedio y tasa de mutación (10 nodos)



Los Gráfico 41 y Gráfico 42 muestran la relación entre el beneficio promedio y densidad, respectivamente, y las generaciones. Se observa que el número de generaciones resultó escaso para hacer converger al modelo teórico.

Gráfico 41. Beneficio promedio y generaciones (10 nodos)

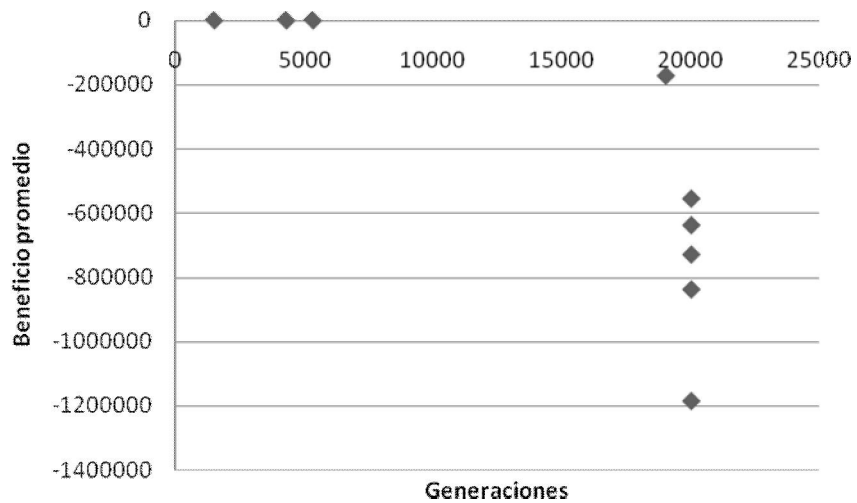
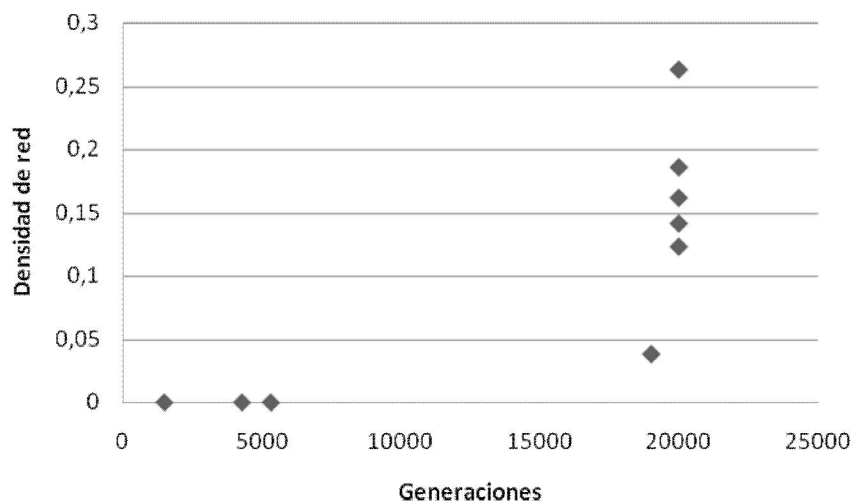


Gráfico 42. Densidad de red promedio y generaciones (10 nodos)



4.2.4.2 Red de 20 nodos

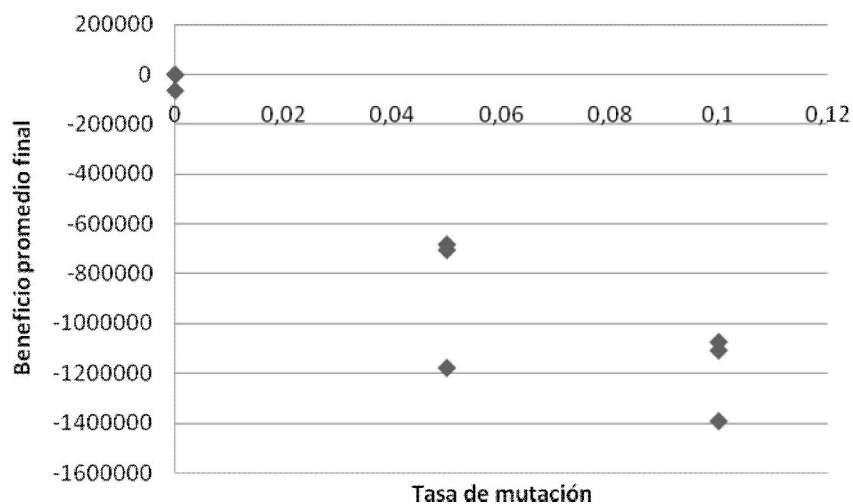
Ahora analizamos el caso de la red de 20 nodos. Sólo con utilizando selección por elitismo se obtiene un valor esperado al teórico (ver Tabla 42).

Tabla 42. Resultados de las simulaciones, 20 nodos

Selección	Mutación	Beneficio encontrado	Densidad encontrada	Ratio acierto	Generaciones
Tournament	0	-64000	0.0168	0.0	20000
Elitism	0	0	0	1.0	2697
Roulette	0	-1054000	0.227	0.0	20000
Tournament	0.05	-682000	0.179	0.0	20000
Elitism	0.05	-704000	0.185	0.0	20000
Roulette	0.05	-1178000	0.31	0.0	20000
Tournament	0.1	-1108000	0.292	0.0	20000
Elitism	0.1	-1074000	0.283	0.0	20000
Roulette	0.1	-1392000	0.367	0.0	20000

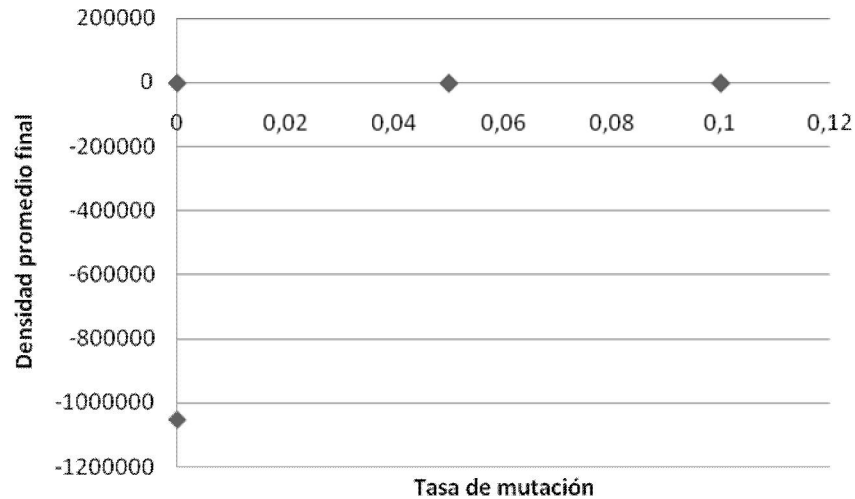
El Gráfico 43 muestra una relación negativa en la tasa de beneficios promedio y la tasa de mutación. Simplemente a mayor mutación, mayor alejamiento del valor teórico.

Gráfico 43. Beneficios promedio y tasa de mutación (20 nodos)



El Gráfico 44 la inexistencia de relación entra la densidad promedio y la tasa de mutación para este ejercicio en particular.

Gráfico 44. Densidad promedio y tasa de mutación (20 nodos)



No se encuentra relación entre la tasa de mutación y ninguno de las variables de desempeño analizadas.

4.2.4.3 Conclusiones parciales

Ambos modelos de colaboración en redes se distinguen de los de formación de redes por trabajar con una función de producción en la cual la creación de enlaces es la que altera el valor de la función creando los incentivos de los agentes hacia mayor o menor cantidad de enlaces. Los resultados de las corridas realizadas muestran una fuerte tendencia a converger en redes densas con ocasionales redes completas. De todos modos, nuevamente aparece el problema de un espacio de búsqueda más amplio con más generaciones. En el caso del ejemplo del modelo de Billand y Bravard (2004) se logró un tímido grado de acierto en la red más pequeña de 10 nodos. Dicho grado de acierto estuvo asociado a una tasa de mutación positiva pero pequeña.

4.3 Conclusiones

Se ejecutaron diversos experimentos computacionales para comprobar las hipótesis teóricas y el grado de acoplamiento observado entre ambos. En ese sentido, se pueden destacar los siguientes hallazgos:

- Con tamaños de red reducidos se encontraron resultados positivos y los modelos teóricos fueron encontrados por el AG.
- Estos hallazgos sólo se circunscriben a las topologías extremas, de red vacía y red completa.
- El número de generaciones adoptado para realizar las corridas (15.000 generaciones) resultó a partir de redes de 10 nodos en una limitante para hacer converger al AG al valor teórico, para todos los modelos.
- La incidencia del proceso de selección puede delimitarse expresando que los procesos de selección basados en torneos demostraron ser más eficientes en las redes dirigidas mientras que el proceso de ruleta fue más eficiente en las redes no dirigidas.
- La tasa de mutación nula se encontró más relacionada a resultados eficientes en el caso de las redes que enfrentan costos directos mientras que una tasa de mutación pequeña fue más eficiente en el caso de redes que enfrentan costos de conexión acumulativos.
- Para los modelos de colaboración en redes, el AG muestra nuevamente problemas asociados a la complejidad NP.

Capítulo 5. Conclusiones e implicaciones

5.1 Introducción

Los algoritmos genéticos han sido la herramienta de evaluación de esta investigación. Se ha utilizado como una técnica de evolución de soluciones a un problema de optimización. Asimismo pueden directamente interpretarse como una metáfora del aprendizaje social. En ese sentido el aprendizaje por algoritmos genéticos debiera reflejar una clase de estabilidad entre medio de la convergencia asintótica y la explosividad. Ello se ha observado a lo largo de todas y cada una de las simulaciones realizadas. Los resultados siempre convergieron o permanecieron estables alrededor de un valor determinado. No se ha detectado conductas explosivas de ningún tipo.

5.1 Conclusiones acerca de las hipótesis de investigación

Las hipótesis de investigación planteadas sólo obtuvieron respuesta positiva para tamaños pequeños de redes siendo el tamaño de red una limitante para hacer afirmaciones respecto a las hipótesis en redes más grandes. Aún para redes pequeñas se encontró respuesta

favorable para las topologías extremas. Dentro del rango de valores intermedios para los casos simulados convergieron a valores de densidad de red alrededor de la media. En ese sentido cabe mencionar la utilidad del procedimiento de selección por ruleta para los grafos no dirigidos y las variantes del torneo en el caso de los grafos dirigidos. Del mismo modo, la tasa de mutación positiva se muestra efectiva en este último caso mientras que la evitación de mutación hace lo propio para el caso de grafo dirigidos.

5.2 Conclusiones acerca de la problemática general de la investigación

El problema de encontrar soluciones evolutivas al juego de formación de redes se encuentra claramente enmarcado en la complejidad NP. Cada nodo agregado a la red eleva exponencialmente el número de alternativas a considerar. Ello se observa claramente en el resultado de este trabajo. Cuando el problema es tratable, los resultados generales esperados asimilables al resultado teórico emergen en gran medida. En ese sentido, este trabajo modeliza computacionalmente por primera vez los modelos expuestos.

5.3 Implicaciones para la teoría

Los resultados presentados muestran la presencia de una evolución estable de las conductas de los agentes siguiendo el esquema de incentivos presentado en cada juego. En ese sentido, los esquemas de pagos que favorecen la desconexión generan estructuras topológicas desconectadas entre todos los individuos y la situación inversa se presenta cuando la estructura de pagos favorece grandemente la conexión social. Este trabajo representa en ese sentido y hasta donde mi conocimiento alcanza, la primera vez que se aplican un esquema de aprendizaje evolutivo al estudio de la eficiencia en juegos de

formación de comunicación monodireccional (otras contribuciones como Carayol y Roux (2005), y Carayol et al. (2005,2008) utilizan sólo grafos no dirigidos).

5.4 Limitaciones

Dentro de las limitaciones de la tesis se deben destacar en primer lugar el escaso número de corridas realizadas para cada caso de estudio. En ese sentido debe destacarse que el entorno de programación NetLogo, si bien ha recibido y recibe permanentemente actualizaciones que mejoran su desempeño, todavía siguen siendo un lenguaje interpretado basado en Java. Ello hace que simulaciones de decenas de corridas con miles de generaciones conlleven tiempos de ejecución de una semana en varios casos.

En segundo lugar, el estudio que se presenta ha sido realizado para valores puntuales que satisfacen las condiciones expuestas en la teoría. No se ha realizado un mapeo completo de valores ni se ha profundizado en el estudio de los valores umbrales que hace que comience a prevalecer una topología sobre otra.

5.5 Futuros caminos de la investigación

Como desprendimiento de este trabajo inicial se planea ampliar el número de generaciones por corrida y de corridas por caso de estudio para realizar más precisas comparaciones estadísticas entre los modelos. Asimismo, realizar una exploración más detallada del rango de valores críticos que hacen eficiente una topología respecto a otra.

Posteriormente se espera explorar modelos de formación de redes más actualizados en los que la información es imperfecta respecto a la estructura de la red (McBride 2006, 2008) de modo de explorar evolutivamente estructuras de red estocásticas.

Apéndice 1. Glosario

Natural	Algoritmo genético
Cromosoma	Cadena de caracteres
Gen	Característica, función o detector
Alelo	Valor de la característica
Lugar	Posición en la cadena de caracteres
Genotipo	Estructura
Fenotipo	Conjunto de parámetros, soluciones alternativas, una estructura decodificada
Epístasis	No linealidad

Fuente: Goldberg (1989: 22)

Apéndice 2. Cálculo de distancias de recorrido de la información

El Seudocódigo 11 muestra el procedimiento propuesto por Floyd y Warshall

Seudocódigo 12. Algoritmo de Floyd-Warshall

```
Algoritmo de Warshall Floyd
1 // Supongamos una función de costos edgeCost(i,j) la cual retorna el costo del enlace de i a j
2 // (infinito si no hay ningún enlace).
3 // También ahora se supone que n es el número de vértices y edgeCost(i,i)=0
4
5 int path[][];
6 // Una matriz bidimensional. En cada paso del algoritmo, path[i][j] es el camino más corto
7 // desde i a j usando los vertices intermedios (1..k-1). Cada path[i][j] es inicializado a
8 // edgeCost(i,j) o infinito si no hay enlace entre i y j.
9
10 procedure FloydWarshall ()
11   for k: = 1 to n
12     for each (i,j) in {1,..,n}2
13       path[i][j] = min ( path[i][j], path[i][k]+path[k][j] );
End
```

El Código Netlogo 13 muestra la correspondiente traducción del Seudocódigo 12 al lenguaje quasi-Pearl del ambiente Netlogo.

Código Netlogo 13. Algoritmo de Floyd-Warshall

```
to warshall
set sum-links 0
;; It creates the reachability-matrix by adding individual's node genome
set adjacency-matrix []
set transitive-matrix []

;; esta es la matriz de adyacencia
let p 0
while [p < num-nodes]
[ set adjacency-matrix lput [genome] of node p adjacency-matrix
  set sum-links sum-links + sum [genome] of node p
  set p p + 1
]

;;; Warshall's algorithm
;;;;;;;;;;;;;
let i 0
let j 0
let k 0

;; Iniciamos con la matriz de clausura transitiva igualándose con la matriz de adyacencia
set transitive-matrix adjacency-matrix
```

```

let track 1

while [i < num-nodes]
  [ set j 0
  while [j < num-nodes]
    [if (item j item i transitive-matrix = 1)
      [set k 0 set track 1
      while [k < num-nodes]
        [ if (item j item k transitive-matrix = 1)
          [set transitive-matrix (replace-item i transitive-matrix (
            replace-item k (item i transitive-matrix) track ))]
          set track track + 1
          set k k + 1]]
        set j j + 1]
      set i i + 1]
;; Finalmente obtenemos la matriz de clausura transitiva de la matrix de adyacencia original

end

```

Apéndice 3. Programación en NetLogo

A1. Introducción

NetLogo es un entorno de modelado programable para simular fenómenos naturales y sociales desde la perspectiva de la modelización basada en agentes. Fue escrito por Uri Wilensky en 1999 (Wilensky, 1999) y ha estado en constante evolución desde entonces en el Center for Connected Learning and Computer-Based Modeling (Centro para el Aprendizaje Conectado y la Modelización informatizada) de la Northwestern University. Es especialmente adecuado para el modelado de sistemas complejos dinámicos. Los creadores de modelos pueden dar instrucciones a los cientos o miles de "agentes" de todos los que operan de forma independiente. Esto hace que sea posible explorar la conexión entre el comportamiento a nivel micro de los individuos y el macro-nivel de los patrones que emergen de la interacción de muchos individuos.

El programa se presenta con una amplia documentación y tutoriales. Es provisto con una biblioteca de modelos, que es una gran colección de simulaciones pre-escritas que pueden ser utilizadas y modificadas. Estas simulaciones abordan muchas áreas de contenido en las ciencias naturales y sociales, incluyendo la biología y la medicina, la física y la química, las matemáticas y ciencias de la computación, y la economía y la psicología social.

Es la generación siguiente de una serie ambientes multi-lenguajes de modelado de agentes que se inició con StarLogo. NetLogo se ejecuta en la máquina virtual Java, así que funciona en todas las principales plataformas (Mac, Windows, Linux, et al). Se ejecuta también como una aplicación independiente. Modelos y actividades HubNet se puede ejecutar como applets de Java en un navegador web.

Dada la teoría expuesta en el segundo capítulo y la descripción presentada en el tercer capítulo sobre los algoritmos genéticos (AG), el presente capítulo plantea la modelización del esquema teórico de juegos de formación de redes en un esquema de AG.

En este capítulo se presentará los experimentos computacionales para verificar como los modelos teóricos son modelados y analizados mediante el esquema de algoritmos genéticos. En ese sentido un esquema general será presentado con variaciones en el esquema de *fitness* el cual será adaptado a cada modelo particular proveyendo esto la adaptación a cada modelo teórico.

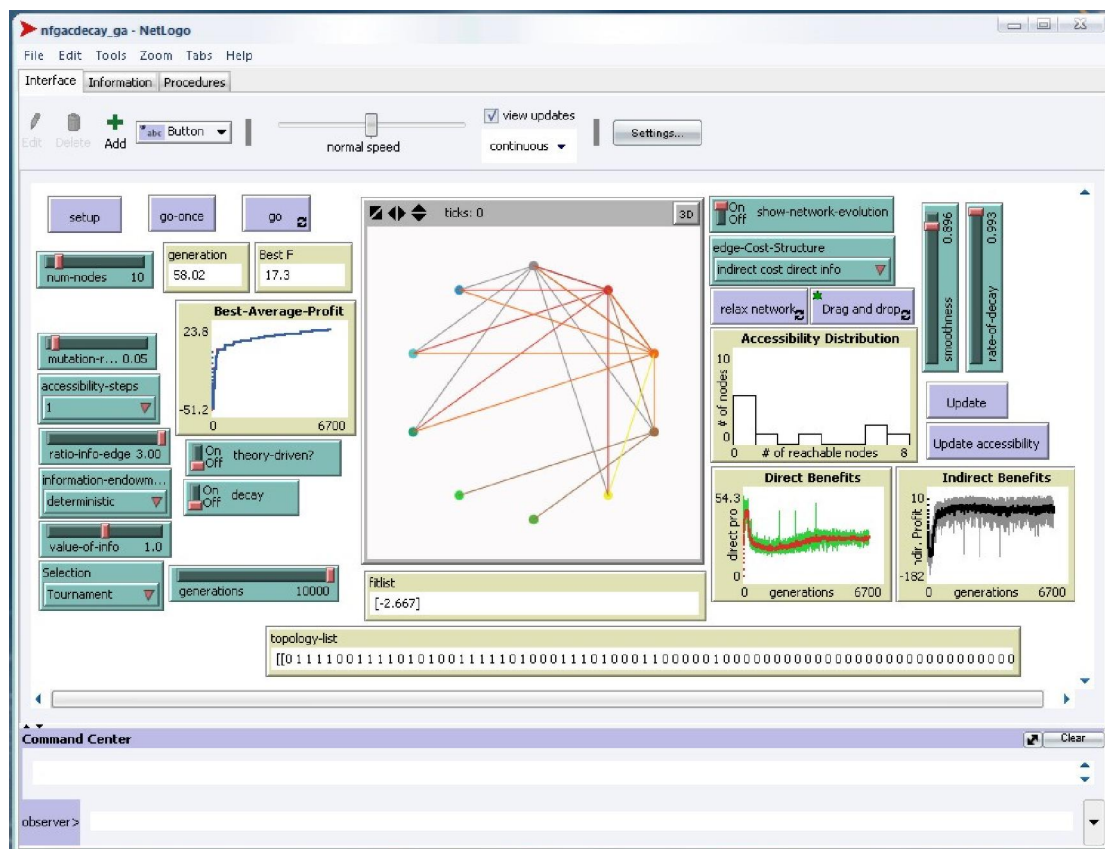
El modelado será desarrollado en un ambiente de programación basada en agentes llamado Netlogo. Ese ambiente de programación ha sido creado en Java e interpreta un lenguaje de programación denominado Quasi-Pearl. En todos los casos analizados se generará una población inicial aleatoria (o una serie de poblaciones) a partir de la cual se valorará su desempeño y mediante diferentes estrategias de selección se extraerán individuos cuales serán apareados para obtener nuevos individuos y así generar una nueva población desarrollándose este procedimiento hasta la activación de algún procedimiento de finalización.

A.2 Esquema general

El código presenta una interface en donde se aprecia una red circular con un mínimo de tres nodos y un máximo de 100, pudiéndose elegir el número de nodos de la que se pretende

analizar a través de un menú deslizante a la derecha. Del mismo modo se presentan diversos menús desplegables, deslizantes y botones de activación para ir seleccionando las diversas configuraciones de algoritmo genético. Entre las opciones se encuentran seleccionar los parámetros de los operadores genéticos así como elegir diversas estructuras de costos de interconexión y recepción de información directa o indirecta. A medida que el algoritmo realiza la búsqueda de las estructuras evolutivas de equilibrio va presentando cada una de las corridas en pantalla. Parte de la información se presenta en menús de la interfaz. El Gráfico 1 muestra la interfaz de uno de los cuatro códigos realizados para este trabajo, siendo todos básicamente similares.

Gráfico 45. Interface



A.3 Principios de programación en ambiente NetLogo

Agentes

El mundo NetLogo se compone de los agentes. Los agentes son seres que pueden seguir las instrucciones. Cada agente puede llevar a cabo su propia actividad, todo simultáneamente.

En NetLogo, hay cuatro tipos de agentes: las tortugas, los parches, los vínculos, y el observador. Las tortugas son los agentes que se mueven alrededor del mundo. El mundo es de dos dimensiones y se divide en una cuadrícula de parches. Cada parche es un cuadrado de "tierra" sobre las que las tortugas pueden moverse. Los enlaces son los agentes que conectan dos tortugas. El observador no tiene un lugar - usted puede imaginarlo cómo mirando al mundo de las tortugas y los parches.

Cuando NetLogo se inicia, no hay tortugas todavía. El observador puede crear nuevas tortugas. Los parches pueden crear nuevas tortugas también. (Los parches no se pueden mover, pero por lo demás son tan "vivos", como las tortugas y el observador son.)

Los enlaces no tienen las coordenadas, sino que se tienen dos extremos (cada una tortuga). Enlaces aparecen entre los dos extremos, a lo largo de la ruta más corta posible, incluso si eso significa envolver todo el mundo.

La forma en que el World o landscape de los parches está conectado puede cambiar. Por defecto, el mundo es un toro que significa que no está acotado, pero "se ajusta" - así que cuando una tortuga se mueve más allá del borde del mundo, desaparece y reaparece en el borde opuesto y cada parche tiene el mismo número de "muestras cercanas", si usted es un parche en el borde del mundo, algunos de sus "vecinos" están en el borde opuesto. Sin embargo, puede cambiar las configuraciones de ajuste con el botón Configuración. Si el envase no está permitido en una dirección dada a continuación, en esa dirección (X o Y) el mundo es limitada. Parches a lo largo de esa frontera que tenga menos de 8 vecinos y las tortugas no se mueve más allá del borde del mundo. Vea la sección de Topología para más información.

Procedimientos

En NetLogo, los comandos y los reporteros les dicen a los agentes qué hacer. Un comando es una acción de un agente para llevar a cabo. Un reportero se obtiene un resultado e informar de ello.

La mayoría de los comandos comienzan con verbos ("create", "die", "jump", "inspect", "clear"), mientras que la mayoría de los reporteros son sustantivos o sintagmas nominales.

Los comandos y los reporteros integrados en NetLogo son llamados primitivas. El Diccionario NetLogo tiene una lista completa de comandos integrados y los reporteros.

Los comandos y los reporteros que se definen a sí mismo se denominan procedimientos. Cada procedimiento tiene un nombre, precedido por la palabra to. La palabra clave end marca el final de los comandos en el procedimiento. Una vez que se define un procedimiento, se puede utilizar en otra parte del programa.

Muchos de los comandos y los reporteros reciben entradas (inputs) - los valores que el comando o el reportero utiliza para llevar a cabo sus acciones.

Ejemplos: Aquí hay dos procedimientos de comandos:

```
to setup
  clear-all      ;; inicializa el mundo a cero
  crt 10         ;; crea 10 nuevas tortugas
end
```

```
to go
  ask turtles
  [ fd 1          ;; todas las tortugas se mueven un paso
    rt random 10  ;; adelante y rotan n pasos al azar
    lt random 10 ]
end
```

Observe el uso del punto y coma para agregar "comentarios" para el programa. Comentarios que su programa más fácil de leer y entender.

En este programa, `setup` y `go` son comandos definidos por el usuario. Por otro lado, `clear-all`, `crt` ("create turtles"), `ask`, `lt` ("left turn"), y `rt` ("right turn") son todos los comandos primitivos.

`random` y `turtles` son reporteros primitivos. `random` toma un solo número como una entrada y reporta un número entero aleatorio que es menor que la de entrada (en este caso, entre 0 y 9). `turtles` reporta el conjunto de agentes (agentes) constituida por todas las tortugas.

`setup` y `go` pueden ser llamados por otros procedimientos o por botones. Muchos modelos de NetLogo tienen el botón que llama una vez a un procedimiento llamado `setup`, (sólo se requiere inicializar los datos al principio) y un botón de llamada permanente que llama a un procedimiento llamado `go`.

En NetLogo, debe especificar qué agentes - tortugas, parches, enlaces, y al observador - van a ejecutar cada comando. (Si no se especifica, el código se ejecuta por el observador.) En el código anterior, el observador utiliza para pedir que el conjunto de todas las tortugas de ejecutar los comandos entre corchetes.

`clear-all` y `crt` sólo se puede ejecutar por el observador. `fd`, por el contrario, sólo puede ser ejecutado por las tortugas. Algunos otros comandos y los reporteros, como conjunto, pueden ser ejecutados por diferentes tipos de agentes.

Aquí están algunas características más avanzadas que puede aprovechar la hora de definir sus propios procedimientos.

Procedimientos con entradas

Sus propios procedimientos pueden tomar entradas, al igual que lo hacen las primitivas. Para crear un procedimiento que acepta entradas, incluye una lista de nombres de entrada entre corchetes después de que el nombre del procedimiento. Por ejemplo:

```
to draw-polygon [num-sides len]
  pen-down
  repeat num-sides
    [ fd len
      rt 360 / num-sides ]
end
```

En el resto del programa, usted podría pedir a las tortugas cada sorteo un octágono con una longitud lateral igual a su número de los que:

```
ask turtles [ draw-polygon 8 who ]
```

Procedimientos de reportero

Al igual que usted puede definir sus propios comandos, usted puede definir sus propios reporteros. Usted debe hacer dos cosas especiales. En primer lugar, use `to-report` en lugar de comenzar el procedimiento con `to`. Luego, en el cuerpo del procedimiento, utilice el reporte para comunicar el valor que desea que se informe.

```
to-report absolute-value [number]
  ifelse number >= 0
    [ report number ]
    [ report (- number) ]
end
```

Variables

Variables de agentes

Agente de variables son los lugares para almacenar los valores (como los números) en un agente. Una variable de agente puede ser una variable global, una variable de tortuga, una variable de parche, o una variable de enlace.

Si una variable es una variable global, hay un solo valor para la variable, y cada agente puede acceder a ella. Usted puede pensar en las variables globales como pertenecientes al observador.

Las variables de tortuga, parche, y enlace son diferentes. Cada tortuga tiene su propio valor para cada variable de tortuga, y cada parche tiene su propio valor para cada variable de parche, y lo mismo para los enlaces.

Algunas variables están integradas en NetLogo. Por ejemplo, todas las tortugas tienen una variable de `color`, y todos los parches tienen una variable de `pcolor`. (La variable parche comienza con "p" para que no se confunda con la variable de la tortuga.) Si se establece la variable, la tortuga o cambios de parches de color.

Otras variables incorporadas tortuga incluyendo `xcor`, `ycor`, y `heading`. Otras variables incorporadas de parches incluyen `pxcor` y `pycor`.

Usted también puede definir sus propias variables. Usted puede hacer una variable global mediante la adición de un interruptor o un control deslizante para su modelo, o con los globales clave en el comienzo de su código, así:

```
globals [ score ]
```

También se puede definir nueva tortuga, el parche y enlace variables usando las tortugas-propias, parches-y vínculos propios, palabras propias, como esto:

```
turtles-own [energy speed]
patches-own [friction]
links-own [strength]
```

Estas variables pueden ser utilizadas libremente en su modelo. Utilice el comando `set` para establecerlos. (Si no se les van a empezar a almacenar un valor de cero.)

Las variables globales se pueden leer y establecer en cualquier momento por cualquier agente. Además, una tortuga puede leer y establecer las variables de revisión de la revisión es de pie. Por ejemplo, este código:

```
ask turtles [ set pcolor red ]
```

hace que cada tortuga para hacer la revisión que está de pie en la luz roja. (Dado que las variables de parches son compartidos por las tortugas de esta manera, no se puede tener una variable de tortuga y una variable de parche con el mismo nombre.)

En otras situaciones en la que desea un agente para leer variable de un agente diferente, usted puede aprovechar. Ejemplo:

```
show [color] of turtle 5
;; imprime el color actual de la tortuga cuyo número who es 5
```

También se puede utilizar de una expresión más complicada que el nombre de una variable, por ejemplo:

```
show [xcor + ycor] of turtle 5
;; imprime la suma de las coordenadas x e y de la
;; tortuga cuyo número who es 5
```

Las variables locales

Una variable local se define y utiliza sólo en el contexto de un determinado procedimiento o parte de un procedimiento. Para crear una variable local, utilice el comando `let`. Usted puede utilizar este comando en cualquier lugar. Si lo usa en la parte superior de un procedimiento, la variable *existe* en todo el procedimiento. Si lo usa dentro de un conjunto de corchetes, por ejemplo dentro de un "ask", entonces existirá sólo dentro de los corchetes.

```
to swap-colors [turtle1 turtle2]
  let temp [color] of turtle1
  ask turtle1 [ set color [color] of turtle2 ]
```

```
ask turtle2 [ set color temp ]
end
```

Listas

En los modelos más simples, cada variable tiene sólo una pieza de información, normalmente un número o una cadena de caracteres. La característica de la lista le permite almacenar múltiples piezas de información en una sola variable mediante la recopilación de las piezas de información en una lista. Cada valor en la lista puede ser cualquier tipo de valor: un número o una cadena de bits, un agente o agentset, o incluso otra lista.

Las listas permiten el conveniente embalaje de la información en NetLogo. Si sus agentes realizar un cálculo repetitivo en múltiples variables, podría ser más fácil tener una lista de variables, en lugar de las variables de número múltiples. Varias primitivas simplifican el proceso de realizar el mismo cálculo para cada valor en una lista.

Listas de constante

Usted puede hacer una lista con sólo poner los valores que desee en la lista entre corchetes, como sigue: `set mylist [2 4 6 8]`. Tenga en cuenta que los valores individuales están separados por espacios. Puede hacer listas que contienen números y cadenas de esta forma, así como listas de listas, por ejemplo, `[[2 4] [3 5]]`.

La lista vacía es escrita por poner nada entre paréntesis, así: `[]`.

Crear listas sobre la marcha

Si desea hacer una lista en la que los valores son determinados por los reporteros, en lugar de ser una serie de constantes, utiliza el reportero `lista`. El reportero de la lista acepta otros dos reporteros, los ejecuta, y los informes de los resultados como una lista.

Si quisiera una lista que contiene dos valores aleatorios, que podría utilizar el siguiente código:

```
set random-list list (random 10) (random 20)
```

Esto conjunto `random-list` para una nueva lista de dos números enteros aleatorios cada vez que se ejecuta.

Para hacer las listas más o menos largo, puede utilizar el reportero `lista` con menos o más de dos entradas, pero para poder hacerlo, deberá adjuntar toda la llamada en paréntesis, por ejemplo:

```
(list random 10)
(list random 10 random 20 random 30)
```

Comandos

Comando	Sintaxis e interpretación
ask	<p><i>ask agentset [commands]</i> <i>ask agent [commands]</i></p> <p>El agente determinado o conjunto de agentes (agentes) ejecuta las órdenes dadas.</p> <p>Ej.</p> <pre>ask turtles [fd 1] ;; Todas las tortugas de avanzar un paso ask patches [set pcolor red] ;; Todos los parches de color rojo ask turtle 4 [rt 90] ;, Sólo la tortuga 4 gira a la derecha</pre> <p>Nota: sólo el observador puede pedir a todas las tortugas o todos los parches. Esto impide que sin darse cuenta que todas las tortugas de pedir a todas las tortugas o todos los parches de pedir a todos los parches, que es un error común de hacer si no somos cuidadosos con la que los agentes se ejecute el código que está escribiendo.</p> <p>Nota: Sólo los agentes que están en el agentset en el momento de pedir comienza ejecutar los comandos.</p>
breed	<p>breed</p> <p>Se trata de una variable integrada de tortuga y de enlace. Lleva a cabo la agentset de todas las tortugas o los enlaces de la misma raza que esta tortuga o un enlace. (Para las tortugas o los enlaces que no tienen ninguna raza en particular, se trata del agentset de todas las tortugas o de todos los enlaces, respectivamente.) Puede establecer esta variable a cambio de una tortuga o enlace de la raza.</p> <p>Ej.</p> <pre>breed [cats cat] breed [dogs dog] ;; código de tortuga: if breed = cats [show "meow!"] set breed dogs show "woof!"</pre>
ca	<p>clear-all</p> <p>Resetea todas las variables globales a cero, y llama a todas las variables integradas calls reset-ticks, clear-turtles, clear-patches, clear-drawing, clear-all-plots, y clear-output para inicialicen todo.</p>
if	<p><i>if condición [commands]</i></p> <p>El reportero debe reportar un valor booleano (verdadero ó falso).</p> <p>Si condición reporta verdadero, se ejecutan los <i>commands</i>.</p> <p>El reportero podría reportar un valor diferente para agentes diferentes, de modo que algunos agentes ejecutarían algunos comandos y otros no.</p> <p>Ej.</p> <pre>if xcor > 0[set color blue] ;; las tortugas en la mitad derecha del landscape se tornan azules</pre>
ifelse	<p><i>ifelse reportero [commands1] [commands2]</i></p> <p>El reportero debe reportar un valor booleano (verdadero o falso).</p>

	<p>Si el reportero reporta verdadero, se ejecutan los <i>commands1</i>.</p> <p>Si el reportero reporta falso, se ejecutan los <i>commands2</i>.</p> <p>El reportero podría reportar un valor diferente para agentes diferentes, de modo que algunos agentes ejecutarían los <i>commands1</i> y otros los <i>commands2</i>.</p> <p>Ej.</p> <pre>ask patches [ifelse pxcor > 0 [set pcolor blue] [set pcolor red]] ;; la mitad izquierda del landscape se torna rojo y ;; la otra mitad se torna azul</pre>
item	<p><i>item índice lista</i> <i>item índice string</i></p> <p>En listas, reporta el valor del ítem en la lista mencionada con el índice mencionado.</p> <p>En las cadenas de caracteres (strings), reporta el carácter en la cadena mencionada en el índice citado.</p> <p>Nótese que los índices comienzan de 0, no de 1. (El primer ítem es el item 0, el segundo ítem es item 1, y así.)</p> <p>Ej.</p> <pre>;; suponga que mylist es [2 4 6 8 10] show item 2 mylist => 6 show item 3 "my-shoe" => "s"</pre>
length	<p><i>length lista</i> <i>length string</i></p> <p>Reporta el número de ítems en una lista dada, ó el número de caracteres en una cadena de caracteres.</p>
let	<p><i>let variable valor</i></p> <p>Crea una nueva variable local y le da un valor dado. Una variable local es una que existe solo dentro de un bloque cerrado de commands.</p> <p>Es para crear la variable, si dentro del bloque de commands se quiere cambiar el valor se debe usar <i>set</i>.</p> <p>Ej:</p> <pre>let prey one-of sheep-here if prey != nobody [ask prey [die]]</pre>
lput	<p><i>lput valor lista</i></p> <p>Añade un valor al final de una lista y reporta la nueva lista.</p> <p>Ej.</p> <pre>;; suponga que mylist es [2 7 10 "Bob"]</pre>

	<pre>set mylist lput 42 mylist ;; mylist ahora queda [2 7 10 "Bob" 42]</pre>
max	<p>max lista</p> <p>Reporta el valor numérico máximo en la lista. Ignora otro tipo de ítems.</p> <p>Ej.</p> <pre>show max [xcor] of turtles ;; Imprime la coordenada x de la Tortuga que está más ;; a la derecha en el landscape</pre>
min	<p>min lista</p> <p>Reporta el valor numérico mínimo en la lista. Ignora otro tipo de ítems.</p> <p>Ej.</p> <pre>show min [xcor] of turtles ;; imprime la menor coordenada x de todas las tortugas</pre>
of	<p>[reporter] of agent [reporter] of agentset</p> <p>Para un agente, reporta el valor del reporter para ese agente (tortuga o parche).</p> <p>Ej.</p> <pre>show [pxcor]of patch 3 5 ;; imprime 3 show [pxcor] of one-of patches ;; imprime el valor pxcor de un parche aleatorio show [who * who] of turtle 5 ;;=> 25 show [count turtles in-radius 3] of patch 0 0 ;; imprime el número de tortugas localizadas dentro de un radio de tres parches del origen</pre> <p>Para un conjunto de agentes (agentset), reporta una lista que contiene el valor del reporter para cada agente en el agentset (en orden aleatorio).</p> <p>Ej.</p> <pre>crt 4 show sort [who] of turtles => [0 1 2 3] show sort [who * who] of turtles => [0 1 4 9]</pre>
precision	<p>precision número lugares</p> <p>Reporta el número redondeado a lugares decimales.</p> <p>Si lugares es negativo, el redondeo toma lugar al izquierda del punto decimal.</p> <p>Ej.</p> <pre>show precision 1.23456789 3 => 1.235 show precision 3834 -3 => 4000</pre>
repeat	<p>repeat número [commands]</p> <p>Corre <i>commands</i> un número de veces.</p> <p>Ej.</p>

	<pre>pd repeat 36 [fd 1 rt 10] ;; la tortuga dibuja un círculo</pre>
replace-item	<p>replace-item índice lista valor replace-item índice string1 string2</p> <p>En una lista, reemplaza un ítem en esa lista. índice es el índice del ítem a ser reemplazado, empezando desde 0. (El sexto ítem en una lista tendría un índice de 5.) Nótese que "replace-item" es usado en conjunción con "set" para cambiar una lista.</p> <p>Del mismo modo para una cadena de caracteres, pero el carácter dado del <i>string1</i> removido y los contenidos del <i>string2</i> son cortados en vez.</p> <pre>show replace-item 2 [2 7 4 5] 15 => [2 7 15 5] show replace-item 1 "cat" "are" => "caret"</pre>
set	<p>set variable valor</p> <p>Fija a la variable un valor dado.</p> <p>Variable puede ser cualquiera de los siguientes:</p> <ul style="list-style-type: none"> • Una variable global declarada usando "globals" • La variable global asociada con un menú deslizable, un switch, un selector, ó un cuadro de ingreso de datos (input box). • Una variable perteneciente a un agente que está siendo llamado • Si el agente llamado es una tortuga, una variable perteneciente al parche debajo de la tortuga. • Una variable local creada por el comando <i>let</i>. • Una entrada al procedure actual. • Una variable especial local (?, ?1, ?2...).
sum	<p>sum lista</p> <p>Reporta la suma de los ítems en la lista.</p> <p>Ej.</p> <pre>show sum [energy] of turtles ;; prints the total of the variable "energy" ;; from all the turtles</pre>
while	<p>while [reporter] [commands]</p> <p>Si el reporter reporta falso, sale del ciclo. En cualquier otro caso corre run commands y lo repite.</p> <p>El reporter podría tener diferente valores para diferentes agentes de modo que algunos podrían correr los commands un número diferente de veces que otros agentes.</p> <p>Ej.</p> <pre>while [any? other turtles-here] [fd 1] ;; la tortuga se mueve hasta que encuentra un parche que ;; no tenga ninguna tortuga encima.</pre>

Referencias

- Airoldi, E.M. (2005). "Sampling algorithms of pure network topologies: Stability and separability of metric embeddings". CMU-ISRI-05- 11, School of Computer Science, Carnegie Mellon University.
- Alemdar, N.M. y S. Özyildirim (1998). "A Genetic Game of Trade, Growth and Externalities". *Journal of Economic Dynamics and Control*: **22**(6): 811-832.
- Alkemade, F. (2007). EVOLUTIONARY AGENT-BASED ECONOMICS. Tesis doctoral, Technische Universiteit Eindhoven, Países Bajos.
- Angus, S. (2006). "Endogenous Communication Networks with Boundedly Rational Agents". Working paper, School of Economics, The University of New South Wales.
- Axelrod, R. (1997). THE COMPLEXITY OF COOPERATION. Princeton, NJ: Princeton University Press.
- Axtell, R.L. (1999). "The Emergence of Firms in a Population of Agents: Local Increasing Returns, Unstable Nash Equilibria, And Power Law Size Distributions". *Working Paper* No. 3. Center on Social and Economic Dynamics. The Brookings Institution.
- _____ (2000). "Why Agents? On the Varied Motivations for Agent Computing in the Social Sciences". Manuscrito. Center on Social and Economic Dynamics, The Brookings Institution.
- _____ (2003). "Effects of Interaction Topology and Activation Regime in Several Multi-Agent Systems". *Working Paper* No. **13**. Center on Social and Economic Dynamics, The Brookings Institution.
- Bala, V. y S. Goyal (2000). "A Noncooperative Model of Network Formation". *Econometrica* **68**: 1181-1229.
- Beal, S. y N. Quérou (2005). "Bounded Rationality and Repeated Network Formation". Working paper, University of Saint-Etienne.

- Berger, T. (2001). "Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis". *Agricultural Economics* **25**: 245-260.
- Berninghaus, S.K., M. Ott y B. Vogt (2004). "Restricting the benefit flow from neighbors: Experiments on network formation". Sonderforschungsbereich **504**. Mannheim Universitat.
- Berninghaus, S.K., K.-M. Ehrhart, M. Ott y B. Vogt (2007). "Evolution of networks—an experimental analysis". *Journal of Evolutionary Economics* **17** (3): 317-347.
- Billand, P. y C. Bravard (2004). "Non-cooperative networks in oligopolies". *International Journal of Industrial Organization* **22**: 593-609.
- _____ (2005). "A Note on the Characterization of Nash Networks". *Mathematical Social Sciences* **49**: 355-365.
- Billand, P., C. Bravard, y S. Sarangi (2008). "Existence of Nash Networks in One-Way Flow Models". *Economic Theory* **37** (3): 491-507(17).
- Birchenhall, C., N. Kastrinos y S. Metcalfe (1997). "Genetic algorithms in evolutionary modelling". *Journal of Evolutionary Economics* **7**: 375-393.
- Birchenhall, C.R. y J-S. Lin (2000). "Learning and Adaptative Artificial Agents: Analysis of an Evolutionary Economic Model". Manuscrito. School of Economic Studies, University of Manchester.
- Brent, E y A. Thompson (1999). "Modeling Social Interactions with Autonomous Agents". *Social Science Computer Review* **17** (3): 313-322.
- Brent, E., A. Thompson, y W. Vale (2000). "A Computational Approach to Sociological Explanations". *Social Science Computer Review* **18** (2): 223-235.
- Blume, L.W. y S.N. Durlauf (2000). "The Interactions-based Approach to Socioeconomic Behavior". *SSRI Working Paper* 2001, University of Wisconsin at Madison, January.
- Cajueiro, D.O. (2005). "Agents preferences and the topology of networks". *Physical Review E* **72**, 047104-1-4.
- Carayol, N. y P. Roux (2005). "A strategic model of complex networks formation". Manuscrito. Université Louis Pasteur – Université Paris Sud.
- Carayol, N., P. Roux y M. Yildizoglu (2005). "Coordination failures in network formation". ACEPOL05 Workshop, ZiF, Bielefeld Universität.

- _____ (2008). "Efficiency of network structures: The needle in the haystack". *Review of Economic Design* 11(4): 339-359.
- Carley, K. (1996). "Validating Computational Models". Manuscrito. Department of Social and Decision Sciences, Carnegie Mellon.
- Cederman, L-E (2003). "Modeling the Self-Organized Criticality of War Size. From Billiard Balls to Sandpiles". *American Political Science Review* 97: 135-150.
- Chen, S-H. (2001). "Fundamental Issues in the Use of Genetic Programming in Agent-Based Computational Economics". Manuscrito. AI-ECON Research Center, Department of Economics, National Chengchi University, Taiwán.
- Chen, S-H. y Ch-Ch. Ni (2000). "Simulating the Ecology of Oligopoly Games with Genetic Algorithms". *Knowledge and Information Systems: An International Journal* (2): 310-339.
- Cohen, M.D.; R.L. Riolo y R. Axelrod (1998). "The Emergence of Social Organization in the Prisoner's Dilemma: How Context-Preservation and others Factors Promote Cooperation". Manuscrito. University of Michigan.
- _____ (2001). "The Role of Social Structure in the Maintenance of Cooperative Regimes". *Rationality and Society* 13(1): 5-32.
- D'Ignazio, A. y E. Giovannetti (2004). "From Exogenous to Endogenous Networks: Internet Applications". CWPE 0445, University of Cambridge.
- Dorsey, R.E. y W.J. Mayer (1992). "Genetic algorithms for estimation problems with multiple optima, non-differentiability and other irregular features". Manuscrito. University of Mississippi.
- Fagiolo, G., M. Valente, y N.J. Vriend (2005). "Segregation in Networks". *Working Paper* 549, Queen Mary, University of London.
- Falk, A. & M. Kosfeld (2003). "It's all about Connections: Evidence on Network Formation," *IZA Discussion Papers* 777, Institute for the Study of Labor (IZA)
- Gen, M., R. Cheng y L. Lin (2008). *NETWORK MODELS AND OPTIMIZATION*. Springer Verlag: London.
- Gilles, R.P. y S. Sarangi (2003). "The Role of Trust in Costly Network Formation". Discussion Paper No. 2003-53, Center for Economic Research, Tilburg University, Mayo.
- Goldberg, D. (1989). *Genetic Algorithms in Search Optimization, and Machine Learning*, New York, Addison-Wesley.

- Goyal, V. y S. Joshi (2003). "Networks of collaboration in oligopoly". *Games and Economic Behavior* **43**: 57-85.
- Goyal, V. (2007). *CONNECTIONS*. Princeton: Princeton University Press.
- Grefenstette, J.J. y J.E. Baker (1994). *How Genetic Algorithms Work: A Critical Look at Implicit Parallelism*. Incluido Buckies, B.P. y F.E. Petry (eds.) *GENETIC ALGORITHMS*. IEEE Computer Society Press: 12-19.
- Gulyás, L. (2002). "On the Transition to Agent-Based Modeling". *Social Science Computer Review* **20** (4): 389-399.
- Iba, T. (1998). "Agent-Based Simulation Model for Bubbles, Crashes and Winner-Take-All Market". Tesis de Maestría, School of Media and Governance, Keio University, Japón.
- Ioannides, Y. M. (2001). "Topologies of Social Interactions". Working Paper 2001-4, Department of Economics, Tufts University, Massachusetts.
- Jackson, M. (2003). "Survey of Network Formation". *Working Paper*, California Institute of Technology.
- Johnson, P.E. (2002). "Agent-Based Modeling. What I've Learned From the Artificial Stock Market". *Social Science Computer Review* **20** (2): 174-186.
- Judd, K.L. (1999). *NUMERICAL METHODS IN ECONOMICS*. Cambridge, MA: The MIT Press. 2nd printing.
- Larrosa, J.M.C. y F. Tohmé (2002). "Network Formation with Heterogeneous Agents". Anales de la XXXVII Reunión Anual de la Asociación Argentina de Economía Política, Tucumán, 13-15 Noviembre.
- _____ (2003a). "Circular Networks as Efficient Nash Equilibria: Two Approaches". XXXVIII Reunión Anual de la Asociación Argentina de Economía Política, Mendoza.
- _____ (2003b). "Formación de redes dirigidas circulares con costos de enlace compartidos". *Estudios Económicos* Vol. XX (N.S.), N° **41**: 27-48.
- _____ (2005). "Redes de equilibrio con costos de conexión acumulativos y comunicación bidireccional" *Estudios Económicos* Vol. XXII (N.S.), N° **44**: 67-88.
- Lieberman, E., C. Hauert y M.A. Nowak (2005). "Evolutionary dynamics on graphs". *Nature* Vol. 433 (20): 312-316.
- McBride, M. (2006). "Imperfect Monitoring in Communication Networks". *Journal of Economic Theory* **126** (1): 97-119.

- _____ (2008). "Position-specific Information in Social Networks". *Mathematical Social Science* **56** (2): 283-295.
- Macy, M.W. y R. Willer (2002). "From Factors to Actors: Computational Sociology and Agent-Based Modeling". *Annual Review of Sociology* **28**: 143-166.
- Macal, C.M. y Sallach, D.L. (2001). "Introduction. The Simulation of Social Agents". *Social Science Computer Review* **19** (3): 245-248.
- Marks, R.E. (1998). "Niche Strategies". Mimeo. Australian Graduate School of Management.
- _____ (2000). "Playing Games with Genetic Algorithms". Manuscrito. Australian Graduate School of Management.
- Mengshoel, O.J. y D.C. Wilkins (1998). "Genetic Algorithms for Belief Network Inference: The Role of Scaling and Niching". Mimeo. Department of Computer Science. University of Illinois, Urbana-Champaign.
- Mitchell, M. (1993). "Genetic Algorithms". Publicado en L. Nadel y D. Stein (eds.), 1992 LECTURES IN COMPLEX SYSTEMS, SFI Studies in the Sciences of Complexity, Vol. V, Addison-Wesley.
- Moretti, S. (2002). "Computer Simulation in Sociology: What Contribution?" *Social Science Computer Review* **20** (1): 43-57.
- Nisan, N., T. Roughgarden, È. Tardos y V.V. Vazirani (2007). ALGORITHMIC GAME THEORY. New York: Cambridge University Press.
- North, M.J. (2001). "Toward Strength and Stability. Agent-Based Modeling of Infrastructure Markets". *Social Science Computer Review* **19** (3): 307-323.
- Özyildirim, S. (1997). "Computing open-loop noncooperative solution in discrete dynamic games". *Journal of Evolutionary Economics* **7**: 23-40.
- Panzarasa, P. y N.R. Jennings (2002). "Social influence, negotiation and cognition". *Simulation Modelling Practice & Theory* **10**: 417-453.
- Pérez Serrada, A. (1996). UNA INTRODUCCIÓN A LA COMPUTACIÓN EVOLUTIVA. Notas de clase.
- Pruijt, H. (2000). "Programming the Evolution of Cooperation". *Social Science Computer Review* **18** (1): 77-81.
- Riechmann, T. (1998). "Genetic Algorithms and Economic Evolution". Diskussionspapier Nr. **219**. Institut für Volkswirtschaftslehre, Abteilung Wachstum und Verteilung, Universität Hannover.

- _____ (1999). "Learning and behavioral stability. An economic interpretation of genetic algorithms". *Journal of Evolutionary Economics* **9**: 225-242.
- _____ (2001). "Genetic algorithm learning and evolutionary games". *Journal of Economic Dynamics & Control* **25**: 1019-1037.
- _____ (2001). *LEARNING IN ECONOMICS*. Heidelberg: Physica Verlag.
- Schelling, T. C. (1978). *MICROMOTIVES AND MACROBEHAVIOR*. New York: Norton.
- Su, M-Ch. y H-T. Chang (1999). "Topology-Preserving Map Formation Achieved with a Genetic Algorithm and its Application in Cluster Analysis". *Tamkang Journal of Science and Technology* **2** (1): 29-36.
- Takadama, K., Y.L. Suematsu, N. Sugimoto, N.E. Nawa y K. Shimohara (2003). "Cross-Element Validation in Multiagent-based Simulation: Switching Learning Mechanisms in Agents". *Journal of Artificial Societies and Social Simulation* **6** (4).
- Tanenbaum, A.S. (1989). *COMPUTER NETWORKS*. Prentice Hall.
- Tardos, É. y T. Wexler (2007). Network Formation Games and the Potential Function Method. Incluido en Nisan et al. (eds) (2007) *ALGORITHMIC GAME THEORY*. Cambridge University Press; New York: capítulo 19.
- Testfatsion, L. (1999). "How Can Economists Get Alife". Manuscrito. Department of Economics, Iowa State University.
- Testfatsion, L. (2000). "Agent-Based Computational Economics: A Brief Guide to the Literature". Manuscrito. Department of Economics, Iowa State University.
- Tedrick, J. (2003). *A COMPARISON OF TWO METHODS OF NETWORK FORMATION: TOP-DOWN AND BOTTOM-UP*. Master Thesis. University of Illinois at Urbana-Champaign.
- Vallée, T. y M. Yildizoglu (2004). "Présentation des algorithmes génétiques et de leurs applications en économie". *Revue d'Economie Politique* **114**: 711-745.
- Vriend, N.J. (2005). "ACE Models of Endogenous Interactions". Manuscrito. Queen Mary, University of London.
- Wang, T-D., C. Fyfe y J.P. Marney (2002). "A Comparison of an Oligopoly Game and the N-person Iterated Prisoner's Dilemma". Manuscrito. Department of Computing and Information Systems. The University of Paisley. UK.
- Watts, A. (2002). "Non-myopic formation of circle networks". *Economic Letters* **74**: 277-282.
- Wilensky, U. (1999). NetLogo, <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

Witkowski, M., B. Neville, J. Pitt (2003), "Agent mediated retailing in the connected local community", *Interacting with Computers* **15**: 5–32.