ARTÍCULOS

Revista de **Filosofía**
ISSN: 0034-8244

EDICIONES COMPLUTENSE

http://dx.doi.org/10.5209/RESF.62028

Autómatas jugando al Dilema del Prisionero iterado

Antonio Benítez[1]

Recibido: 13 de septiembre de 2016 / Aceptado: 23 de marzo de 2017

Resumen. Este estudio se ocupa de estrategias deterministas para jugar al Dilema del Prisionero iterado. Cada estrategia se incorpora a la tabla de un autómata de estado finito. Se estudian exhaustivamente tanto las estrategias de 4 bits como las de 16 bits. El estudio de las estrategias de 64 bits se ha hecho por medio de un Algoritmo Genético. Tanto la idea de estudiar estrategias deterministas como la de servirse de un Algoritmo Genético está en Axelrod, *The Complexity of Cooperation*.

Respecto a las estrategias de 4 bits, una conclusión se ha impuesto: *Tit for Tat* no es la estrategia ganadora.

Respecto a las estrategias de 16 bits, tampoco *Tit for Tat* ha resultado ganadora.

Respecto a las estrategias de 64 bits, hay que destacar el método de hacer evolucionar los cromosomas encontrados por el Algoritmo Genético confrontándolos con un conjunto de control de estrategias. También hay estrategias de 64 bits mejores que *Tit for Tat*.
Palabras clave: dilema del prisionero iterado; métodos con estrategias deterministas; torneo computerizado; filosofía experimental.

[en] Automata playing iterated Prisoner's Dilemma

Abstract. This study deals with deterministic strategies for playing the iterated Prisoner's Dilemma. Each strategy is incorporated into the table of a finite state automaton. 4-bit and 16-bit strategies are exhaustively studied on this paper, meanwhile 64-bit strategies have been approached by means of a Genetic Algorithm. Both ideas, of studying deterministic strategies and using a Genetic Algorithm, are in Axelrod's *The Complexity of Cooperation*.

Regarding 4-bit strategies, a conclusion has been imposed: *Tit for Tat* is not the winning strategy.

Regarding 16-bit strategies, *Tit for Tat* has not been the winner either.

Regarding 64-bit strategies, we should mention the method to evolve chromosomes found by the Genetic Algorithm confronting them with a control set of strategies. There are also 64-bit strategies better than *Tit for Tat*.
Keywords: Iterated Prisoner's Dilemma, Methods for Deterministic Strategies, Computer Tournament, Experimental Philosophy.

Sumario. 1. Dilema del prisionero; 1.1. Dilema del prisionero iterado; 1.2. Estrategias deterministas; 1.3. Un lenguaje para representar el problema; 1.4. Longitud de estrategias; 1.5. Los agentes como autómatas de estado finito; 2. Torneo con una situación inicial; 2.1. Resultados computacionales. Opción A; 2.2 Conclusiones; 2.3. Resultados computacionales. Opción B; 2.4. Conclusiones; 3. Torneo con dos situaciones iniciales; 3.1. Resultados computacionales; 3.2. Conclusiones; 4. Torneo

[1] Universidad Complutense de Madrid
abenitez@ucm.es

con tres situaciones iniciales; 4.1. Parametrización del AG; 4.2. Determinación del grado de adaptación (*Fitness*); 4.3. *Fitness* mediante un torneo entre generaciones; 4.4. Resultados computacionales; 4.5. *Fitness* mediante un conjunto de control; 4.6. Evolución del conjunto de control; 4.7. Torneo entre conjuntos de control; 4.8. Torneo entre cromosomas encontrados por el AG; 5. Conclusiones; 6. Referencias bibliográficas.

Cómo citar: Benítez, A. (2018) "Autómatas jugando al Dilema del Prisionero iterado", en *Revista de Filosofía* 43 (2), 223-243.

1. Dilema del prisionero

Es usual presentar intuitivamente el llamado Dilema del prisionero así: la policía arresta a dos individuos –llamemos "A" a uno, y "B" al otro– por haber cometido juntos un crimen. La policía los mantiene en celdas separadas y ofrece a cada uno la libertad si hace una declaración inculpando al otro. La separación entre los presos tiene como consecuencia que no pueden comunicarse entre sí y tampoco saber si el otro ha declarado, y menos aún qué ha declarado.

Cada preso conoce su situación legal. La situación legal de los dos presos se entiende mejor si se considera qué pasará con ellos según que declaren contra el otro o no. Pueden pasar tres cosas:

1. Uno declara contra el otro y este no declara contra el primero. Entonces este segundo será condenado a cadena perpetua y el primero será liberado.
2. Ambos declaran contra el otro. Entonces ambos serán condenados a cárcel pero un número de años no muy grande, sea diez años.
3. Ambos se niegan a declarar. Entonces ambos serán condenados a cárcel pero un número de años reducido, sea dos años.

Si ahora pensamos en qué pasaría si ambos presos pudieran coincidir en la misma celda y hablar, probablemente se dirían lo siguiente: «declarando uno contra el otro es posible quedar en libertad, pero si no llegamos a un trato, ambos iremos a declarar contra el otro y nos caerán diez años. Como ninguno de los dos somos estúpidos, no nos fiáremos de no declarar hasta no ver que en la declaración firmada por el otro no se nos acusa. Por tanto, no hay modo de escapar a la cárcel. Lo mejor, porque entraña la pena menor es negarse a declarar. ¿Estás de acuerdo?», dice uno.

1.1. Dilema del prisionero iterado

La cosa cambia si cambiamos la escena. Seguiremos suponiendo dos actores, cada uno con sus intereses y, por ello, razonablemente egoísta. Pero lo que más cambia es que el drama no es estático sino dinámico, es decir, la situación cambia y se conservan las decisiones tomadas, al menos, en la memoria de cada uno de los actores. Sigue habiendo dos comportamientos posibles en cada situación: «cooperar» y «no cooperar», no declarar o declarar contra el otro respectivamente. Pero a cada situación producida por las decisiones de los dos actores le sigue otra un cierto número de veces. ¿Es posible actuar con una estrategia prefijada, esto es, siguiendo reglas que necesariamente dirán relación a las decisiones tomadas en el pasado? ¿Existe una estrategia ganadora?

1.2. Estrategias deterministas

Una estrategia es determinista si, y solo si, ante una misma configuración de una serie dada de situaciones previas produce la misma decisión. Es el caso de la conocida estrategia *Tit for Tat*, que se reduce a dos reglas:

- R1: si se cumple que es la situación inicial del juego, decidir cooperar.
- R2: en cualquier otra situación, decidir hacer lo mismo que hizo el contrario en la situación precedente.

Simbólicamente:

- R1: $n = 1 \Rightarrow 1$

[para n = número de iteración]

- R2: else ⇒ (last (first serie-iteraciones))

[para *serie-iteraciones* = ((1 1) (0 1) (1 0)…)]

presente ⟶ pasado

Una estrategia no determinista es la que propuso Johann Joss para el primer torneo computerizado que llevó a cabo Axelrod.

- R1: si $n = 1$, decidir cooperar.
- R2: si el contrario no cooperó en la situación precedente, decidir no cooperar.
- R3: si el contrario cooperó en la situación precedente, decidir cooperar nueve de cada diez veces.

La regla R3 es no determinista, no porque suponga una cierta probabilidad (9/10) sino por la manera de aplicar esa probabilidad. Hay que pensar que de diez veces que el contrario coopere, Joss deja de cooperar en una, entonces esa una ha de ser una cualquiera de las diez. Por tanto, no ha de ser la misma cada vez que se cumple que el contrario ha cooperado diez veces.

Esto se ve claro si se hace jugar al autómata cuya estrategia de 4 bits es (1 1 1 1) –cooperante– contra Joss en un encuentro con 11 jugadas. Un resultado es:

Ejecución 1ª

jugada	(1 1 1 1)	Joss	jugada al azar en que se aplicará no cooperar
0	1	1	
1	1	1	
2	1	1	
3	1	0	coop-otro 3
4	1	1	
5	1	1	
6	1	1	
7	1	1	
8	1	1	
9	1	1	
10	1	1	

Ejecución 2ª

jugada	(1 1 1 1)	Joss	jugada al azar en que se aplicará no cooperar
0	1	1	
1	1	1	
2	1	1	
3	1	1	
4	1	1	
5	1	1	
6	1	1	
7	1	0	coop-otro 7
8	1	1	
9	1	1	
10	1	1	

En la iteración inicial ambos contrincantes cooperan. En la iteración 1, dado que el contrario ha cooperado en la iteración 0, el contador de cooperaciones del contrario que Joss tiene se actualiza y a la vez se determina –mediante el generador de números aleatorios– en cuál de las cooperaciones del otro, entre la 1 y la 10, se aplicará la probabilidad 1/10 de no cooperar.

Es claro que dos ejecuciones distintas del mismo enfrentamiento no tienen por qué coincidir en el número de jugada en que se aplique la no cooperación.

Este estudio, salvo que se diga lo contrario, se centrará en estrategias deterministas.

1.3. Un lenguaje para representar el problema

- Signos: {1,0}
  Leyenda:
    - 1: cooperar, o no declarar

- 0: no cooperar, o declarar contra el otro

- Representación de una situación cualquiera

decisión del actor *A*	decisión del actor *B*
1 o 0	1 o 0

Ejemplos:

- (1 1), ambos han cooperado
- (0 1), el primero no coopera y el segundo sí

- El número de situaciones distintas es:

decisión de *A*	decisión de *B*
1	1
1	0
0	1
0	0

Es decir, el número de combinaciones, o situaciones, posibles es $2^2 = 4$.

- Es necesario advertir que cualquier situación, ya sea la única en que dos agentes toman una decisión ya sea que cada situación se inscriba en un largo curso de interacciones o secuencia de situaciones, será una, y solo una, de las determinadas por la combinatoria anteriormente reseñada.
- Puntuaciones obtenidas por cada uno de los dos actores en cada situación posible

Situación			
A	*B*	Puntos *A*	Puntos *B*
0	1	5	0
1	0	0	5
1	1	3	3
0	0	1	1

Tabla 1: Criterio *DP*

- Repárese en que las puntuaciones establecidas en la tabla son arbitrarias, pero cumplen algo esencial: caso de que uno coopere y el otro no coopere, este ultimo se lleva la máxima puntuación porque quedaría en libertad y el que sí ha cooperado obtendría cero puntos porque recibiría cadena perpetua. Además las puntuaciones otorgadas tanto en el caso de la cooperación mutua como en el caso de la mutua no cooperación han de ser menores que la otorgada como puntuación máxima en el primer caso. Y, por último, la puntuación obtenido por no cooperar ambos ha de ser menor que la obtenida por cooperar ambos. Es decir, es posible otra tabla de puntuaciones, por ejemplo:

Situación			
A	*B*	Puntos *A*	Puntos *B*
0	1	10	0
1	0	0	10
1	1	5	5
0	0	3	3

He adoptado la tabla primera porque es la que más suele aparecer en la literatura sobre el Dilema del prisionero.2

Podemos considerar dos agentes que interaccionan entre sí a lo largo de bastante tiempo. El número de interacciones puede ser muy grande, pero no infinito. Simbólicamente podemos adoptar las representaciones siguientes:

- La decisión de un agente en un instante *t* del tiempo: D_t^A o D_t^B
- La estrategia seguida por un agente: E^A o E^B
- Situación actual, determinada por las decisiones de los agentes: $S_t = (D_t^A \; D_t^B)$
- Situación precedente: $S_{t-1} = (D_{t-1}^A \; D_{t-1}^B)$

La decisión de un agente es una función cuyos argumentos son una o más situaciones precedentes y su estrategia:

- Serie de situaciones: S_{t-1}^{t-n}, una serie consecutiva de situaciones precedentes desde la anterior, S_{t-1}, a la más alejada en el tiempo, S_{t-n}
- $D_t^A = f(S_{t-1}^{t-n}, E^A)$
- $D_t^B = f(S_{t-1}^{t-n}, E^B)$

1.4. Longitud de estrategias

Es fácil advertir que la última fórmula, aquella en la que la decisión actual de un agente es función de una serie de situaciones previas más la estrategia que siempre sigue, da paso a tipos distintos de agentes según sea la longitud de la serie de situaciones previas a las que atiende para tomar su decisión. Por ejemplo, si el agente sigue la estrategia conocida como *Tit for Tat*, entonces solo atiende a *una* situación previa. La fórmula se transforma, por tanto, en:

- $D_t^{TfT} = f(S_{t-1}^{t-1}, E^{TfT}) = f(S_{t-1}, E^{TfT})$

Ahora bien, si este agente atiende exclusivamente a la situación previa para tomar su decisión actual, entonces su estrategia habrá de tener una longitud de 4 bits, puesto que las combinaciones posibles de esa situación previa son 2^2. Resulta evidente, pues, que *Tit for Tat* es una de las 16 posibles estrategias de 4 bits.

Si en lugar de atender a una situación previa, el agente atiende a *dos* situaciones previas seguidas, entonces el argumento S_{t-1}^{t-n} de la función f tendrá una longitud de 4 bits. Y el número de combinaciones posibles de 4 bits es $2^4 = 16$. Y

2 Axelrod, R. (1984): *The Evolution of Cooperation*. N. York, Basic Books Inc. Publisher, pág. 8. Mitchell, M. (2009): *Complexity. A Guided Tour*. N. York, Oxford University Press,, pág. 213. Binmore, K. (2007): *Game Theory. A Very Short Introduction*. Oxford, Oxford U. Press, pág. 18, propone puntuaciones alternativas.

por ello, la estrategia de cualquier agente que tenga en cuenta las dos situaciones previas para tomar su decisión habrá de tener una longitud de 16 bits.

Mientras que el número de estrategias distintas de 4 bits es 16, el número de estrategias distintas de 16 bits es $2^{16} = 65.536$.

El mismo razonamiento se aplica al caso de agentes que consideren *tres* situaciones previas. La longitud de cada serie de tres situaciones previas es de 6 bits, por lo que el número de series distintas de 3 situaciones es $2^6 = 64$. Y por ello, una estrategia cualquiera adaptada a una serie de tres situaciones previas habrá de tener una longitud de 64 bits.

El número de posibles estrategias distintas de 64 bits es

$$2^{64} = 18_3 446.744_2 073.709_1 551.616$$ [3]

1.5. Los agentes como autómatas de estado finito

Un agente cuya estrategia tenga una longitud de 4 bits es un agente que atiende exclusivamente a la situación precedente para tomar su decisión actual. Por tanto, este agente puede ser definido formalmente como un *autómata de estado finito*. En efecto, la situación precedente será una de las cuatro siguientes:

1	1
1	0
0	1
0	0

Y ante cada una de ellas el agente tomará siempre la misma decisión. Supongamos que *Tit for Tat* compite contra otro agente de nombre *B*, y representemos en una tercera columna las decisiones que ha de tomar *Tit for Tat*:

TfT_{t-1}	B_{t-1}	TfT_t
1	1	1
1	0	0
0	1	1
0	0	0

Es claro que la tabla anterior es la tabla de un autómata de estado finito, donde la primera columna corresponde al estado en que está el autómata *TfT* la segunda al *input* recibido –decisión anterior de *B*– y la tercera al nuevo estado en que queda el autómata.

El mismo razonamiento cabe hacer para estrategias deterministas de 16 bits y

---

3 Estoy de acuerdo con Axelrod (Axelrod 1997, 18) en que la cifra es tan grande que es imposible intentar una computación exhaustiva. Curiosamente la cifra es la misma que la de las Torres de Hanoi. En un torneo de computadora tipo liga, cada estrategia habría de realizar $(2\times 2^{64}) - 2 = 36_3 893.488_2 147.419_1 103.230$ juegos o partidos. Supuesto 1 segundo por partido, tendríamos unos $1.169_2 884.834_1 710.144$ años, frente a los $20.000_1 000.000$ años que se calcula que tiene el Universo.

64 bits. El caso de estrategias de 16 bits es más fácil de ver puesto que las combinaciones de dos situaciones son 16:

TfT_{t-1}	B_{t-1}	TfT_{t-2}	B_{t-2}	TfT_t
1	1	1	1	1
1	1	1	0	1
1	1	0	1	1
1	1	0	0	1
1	0	1	1	0
1	0	1	0	0
1	0	0	1	0
1	0	0	0	0
0	1	1	1	1
0	1	1	0	1
0	1	0	1	1
0	1	0	0	1
0	0	1	1	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	0

La siguiente tabla quiere dar idea del caso de las estrategias de 64 bits.

Nro Combi-nación	TfT_{t-1}	B_{t-1}	TfT_{t-2}	B_{t-2}	TfT_{t-3}	B_{t-3}	TfT_t
1	1	1	1	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
16	1	1	–	–	–	–	1
17	1	0	–	–	–	–	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
32	1	0	–	–	–	–	0
33	0	1	1	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
48	0	1	–	–	–	–	1
49	0	0	–	–	–	–	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
64	0	0	–	–	–	–	0

Lo que sigue es un estudio de torneos de autómatas de estado finito. En primer lugar, de autómatas correspondientes a estrategias de 4 bits. A continuación, de autómatas con estrategias de 16 bits. Y finalmente con estrategias de 64 bits.

2. Torneo con una situación inicial

Los 16 autómatas corresponden a las 16 estrategias posibles. Cada autómata juega 30 encuentros. Cada encuentro tiene 200 iteraciones.

2.1. Resultados computacionales. Opción A

En cada encuentro, la primera iteración no está predefinida. Cada autómata toma la decisión que corresponde al cambio de estado de la primera fila de su tabla.

orden	autómata	estrategia	puntos DP	30 partidas de 200 jugadas	Promedio p.DP/6000	ganadas	empatadas	perdidas	Puntos Liga
1	12	(0 1 0 0)	18800	6000	3.1333333333333333	24	6	0	54
2	16	(0 0 0 0)	18800	6000	3.1333333333333333	24	6	0	54
3	8	(1 0 0 0)	17952	6000	2.992	10	16	4	36
4	15	(0 0 0 1)	16990	6000	2.8316666666666666	16	6	8	38
5	7	(1 0 0 1)	15542	6000	2.590333333333333	4	18	8	26
6	6	(1 0 1 0)	15022	6000	2.5036666666666667	0	24	6	24
7	14	(0 0 1 0)	15014	6000	2.5023333333333335	16	14	0	46
8	11	(0 1 0 1)	14776	6000	2.462666666666667	12	8	10	32
9	13	(0 0 1 1)	14200	6000	2.3666666666666667	8	14	8	30
10	5	(1 0 1 1)	13992	6000	2.332	0	22	8	22
11	10	(0 1 1 0)	13064	6000	2.1773333333333333	12	12	6	36
12	9	(0 1 1 1)	12016	6000	2.002666666666667	8	10	12	26
13	1	(1 1 1 1)	10800	6000	1.8	0	14	16	14
14	2	(1 1 1 0)	10800	6000	1.8	0	14	16	14
15	3	(1 1 0 1)	10800	6000	1.8	0	14	16	14
16	4	(1 1 0 0)	10800	6000	1.8	0	14	16	14

Las columnas de título «ganadas», «empatadas», «perdidas» y «puntos liga» se refieren a encuentros ganados (2 puntos por cada uno), empatados (1 punto), perdidos (0 puntos) y total de puntos obtenidos. Ganar, empatar o perder se determina según la puntuación DP obtenida en el encuentro.

2.2. Conclusiones

1. La tabla anterior se comenta sola. Los ganadores *ex aequo* son el autómata identificado como 12 y el identificado como 16. Obtienen igual promedio DP. Y también ganan y empatan el mismo número de partidas. Es llamativo que no pierden ningún encuentro.
2. Las tablas de los autómatas 12 y 16 son las siguientes:

Tabla autómata 12

$A\text{-}12_{t-1}$	B_{t-1}	$A\text{-}12_t$
1	1	0
1	0	1
0	1	0
0	0	0

Tabla autómata 16

$A\text{-}16_{t-1}$	B_{t-1}	$A\text{-}16_t$
1	1	0
1	0	0
0	1	0
0	0	0

3. Las estrategias correspondientes a esas tablas pueden definirse así:

Estrategia autómata 12

R1: $S_{t-1} = (1\ 0) \Rightarrow 1$

R2: else $\Rightarrow 0$

Estrategia autómata 16

R1: *true* $\Rightarrow 0$

4. *Tit for Tat* –autómata 6, (1 0 1 0)– se desempeña bastante mal, tanto desde el punto de vista del promedio DP que obtiene –hay una diferencia con los primeros de 0.6296– como desde el punto de vista de encuentros ganados, empatados o perdidos.
   Hay que notar que *Tit for Tat* no gana ningún encuentro, pierde 6 de 30 (1 de cada 5), aunque lo más llamativo es que empata 4 de cada 5 encuentros. El empate se produce tanto cuando ambos agentes no cooperan como cuando ambos sí cooperan.

2.3. Resultados computacionales. Opción B

El número de jugadas de cada encuentro es 200. Cada encuentro se divide en 4 series de 50 jugadas. La primera iteración está predefinida, aunque no se incorpora a la serie de iteraciones: es una de las cuatro posibles. Se hace así para que cada autómata pueda tomar una decisión inicial según su tabla. La puntuación total es la suma de las obtenidas en cada serie de 50 iteraciones. Por tanto, el número de jugadas realizadas para alcanzar esa puntuación es 4×50=200.

orden	autómata	estrategia	puntos DP	30 partidas de 200 jugadas	promedio p.DP/6000	ganadas	empatadas	perdidas	Puntos Liga
1	16	(0 0 0 0)	18800	6000	3.1333333333333333	30	0	0	60
2	8	(1 0 0 0)	18552	6000	3.092	28	0	2	56
3	15	(0 0 0 1)	17400	6000	2.9	22	0	8	44
4	7	(1 0 0 1)	16176	6000	2.696	12	10	8	34
5	12	(0 1 0 0)	16100	6000	2.683333333333333	24	2	4	50
6	14	(0 0 1 0)	13980	6000	2.33	18	8	4	44
7	11	(0 1 0 1)	13924	6000	2.320666666666667	10	10	10	30
8	4	(1 1 0 0)	13500	6000	2.25	10	10	10	30
9	13	(0 0 1 1)	13500	6000	2.25	10	10	10	30
10	6	(1 0 1 0)	13076	6000	2.179333333333333	8	14	8	30
11	5	(1 0 1 1)	12580	6000	2.0966666666666667	4	8	18	16
12	3	(1 1 0 1)	11292	6000	1.882	4	2	24	10
13	10	(0 1 1 0)	11024	6000	1.8373333333333333	8	10	12	26
14	9	(0 1 1 1)	10240	6000	1.7066666666666668	8	0	22	16
15	1	(1 1 1 1)	8400	6000	1.4	0	0	30	0
16	2	(1 1 1 0)	8256	6000	1.376	2	0	28	4

2.4. Conclusiones

1. También en este caso es fácil comentar la tabla anterior. El ganador es el autómata identificado como 16. Obtiene el mejor promedio DP: 3.13, como en la opción A. Y también gana todos los encuentros.
2. *Tit for Tat* –autómata 6, (1 0 1 0)– se desempeña bastante mal desde el punto de vista del promedio DP que obtiene –hay una diferencia con el primero de 0.9540–. Sin embargo, gana 8 encuentros (en la opción A, 0), empata 14 (en la opción A, 24) y pierde 8 (en la opción A, 6).

Gana a los autómatas: 1, 2, 3, 9, empata contra los autómatas: 4, 5, 7, 10, 11, 13, 14 y pierde contra: 7, 8, 12, 15, 16.

3. Torneo con dos situaciones iniciales

Los autómatas suman 65.536, que corresponden a las 65.536 estrategias distintas de 16 bits. Cada autómata juega (65536−1)+(65536−1)= 131.070 partidas o encuentros. Cada encuentro es de 200 iteraciones. La primera iteración no está predefinida y, como antes, cada autómata toma la decisión que corresponde al cambio de estado de la primera fila de su tabla.

3.1. Resultados computacionales

- Cada autómata ha jugado (65536−1) como jugador *A* y (65536−1) como jugador *B*. Total =131070 encuentros o partidas. Como cada partida es de 200 jugadas o iteraciones, cada autómata ha jugado $26_{1}214.000$ iteraciones.

orden	autómata	puntos DP	131070 partidas de 200 jugadas	Promedio p.DP/26.214.000
1	62719	84207720	26214000	3.212318608377203
2	63231	84207720	26214000	3.212318608377203
3	64767	84207720	26214000	3.212318608377203
4	65279	84207720	26214000	3.212318608377203
5	45567	84109672	26214000	3.2085783169298847
6	46079	84109672	26214000	3.2085783169298847
7	46591	84109672	26214000	3.2085783169298847
8	47103	84109672	26214000	3.2085783169298847
9	47615	84109672	26214000	3.2085783169298847
10	48127	84109672	26214000	3.2085783169298847
11	48639	84109672	26214000	3.2085783169298847
12	49151	84109672	26214000	3.2085783169298847
13	61951	84109672	26214000	3.2085783169298847
14	62463	84109672	26214000	3.2085783169298847
15	62975	84109672	26214000	3.2085783169298847
16	63487	84109672	26214000	3.2085783169298847
17	63999	84109672	26214000	3.2085783169298847
18	64511	84109672	26214000	3.2085783169298847
19	65023	84109672	26214000	3.2085783169298847
20	65535	84109672	26214000	3.2085783169298847
20378	3856	67151696	26214000	2.561672999160754

- El autómata 3856 corresponde a la estrategia de *Tit for Tat*. (Véase apartado 1.5.)

orden	autómata	estrategia	Promedio p.DP/26.214.000
1	62719	(0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1)	3.212318608377203
2	63231	(0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1)	3.212318608377203
3	64767	(0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1)	3.212318608377203
4	65279	(0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1)	3.212318608377203
5	45567	(0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
6	46079	(0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
7	46591	(0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
8	47103	(0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
9	47615	(0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
10	48127	(0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
11	48639	(0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
12	49151	(0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
13	61951	(0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
14	62463	(0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
15	62975	(0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
16	63487	(0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
17	63999	(0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
18	64511	(0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
19	65023	(0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
20	65535	(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1)	3.2085783169298847
20378	3856	(1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0)	2.561672999160754

- Finalmente la clasificación atendiendo a los resultados tipo liga.

orden	autómata	ganadas	empatadas	perdidas	Puntos Liga
1	62719	104960	12030	14080	221950
2	63231	104960	12030	14080	221950
3	64767	104960	12030	14080	221950
4	65279	104960	12030	14080	221950
5	45567	107008	15358	8704	229374
6	46079	107008	15358	8704	229374
7	46591	107008	15358	8704	229374
8	47103	107008	15358	8704	229374
9	47615	107008	15358	8704	229374
10	48127	107008	15358	8704	229374
11	48639	107008	15358	8704	229374
12	49151	107008	15358	8704	229374
13	61951	107008	15358	8704	229374
14	62463	107008	15358	8704	229374
15	62975	107008	15358	8704	229374
16	63487	107008	15358	8704	229374
17	63999	107008	15358	8704	229374
18	64511	107008	15358	8704	229374
19	65023	107008	15358	8704	229374
20	65535	107008	15358	8704	229374
20378	3856	0	65534	65536	65534

3.2. Conclusiones

1. *Tit for Tat* no es la estrategia ganadora.
2. *Tit for Tat* queda al final del primer tercio de la tabla de clasificación.
3. Entre el primero y él hay una diferencia de 0,6506 en el promedio.

4. Torneo con tres situaciones iniciales

Como hemos visto más arriba (apartado 1.4.) el número total de autómatas que corresponden a estrategias de 64 bits es $2^{64} = 18_3446.744_2073.709_1551.616$. Ya he dicho que estoy de acuerdo con Axelrod cuando afirma que es prácticamente imposible hacer un torneo en que se enfrenten todos contra todos. Una computación exhaustiva es impensable. Es necesario adoptar otra estrategia.

Axelrod propuso valerse de los Algoritmos Genéticos. Como es sabido, un AG permite una exploración no exhaustiva de un espacio de combinaciones que puede ser muy grande en número. Es el caso de las estrategias de 64 bits.

Voy a presentar a continuación el estudio realizado de estrategias de 64 bits por medio de un AG.

4.1. Parametrización del AG

- Tamaño del individuo: 64
- Tamaño de la población: 40
- Porcentaje de la población para cruzar: 1/2 (los 20 mejores)
- Punto de cruce: al azar
- Probabilidad mutación: 1/1000

4.2. Determinación del grado de adaptación (*Fitness*)

Una vez generada una población inicial, se hace necesario calcular el grado de adaptación (*fitness*) a la tarea para la que han de servir de cada uno de los individuos componentes de dicha población. En nuestro caso, la tarea para la que han de servir es participar en un torneo computerizado enfrentándose a otras estrategias. Por ello, para calcular el grado de adaptación de cada cromosoma de la población entiendo que hay que someterlos a un torneo computerizado. He adoptado dos perspectivas.

4.3. *Fitness* mediante un torneo entre generaciones

Una es llevar a cabo un torneo computerizado con los cromosomas de la población. Al iniciar un ciclo del AG la población generada tiene un tamaño de cuarenta cromosomas. Se lleva a cabo un torneo entre estos cuarenta cromosomas para determinar el *fitness* de cada uno: el número de puntos DP obtenidos en el torneo. Se ordenan y se genera la primera nueva generación. Para determinar el *fitness* de cada cromosoma de la nueva generación se hace un torneo de estos cromosomas enfrentados a ellos mismos más la población preexistente. Se ordena la población temporal, suma de la preexistente más la nueva generación, y se eligen los cuarenta mejores. A continuación se repite el ciclo del AG.

4.4. Resultados computacionales

He realizado 200 aplicaciones de un ciclo del AG –cada ciclo es de 1000 iteraciones–. He acumulado en un archivo los cromosomas de cada ciclo que no existieran ya en el archivo. Al final he contado con 1936 cromosomas.

1. Estos 1936 cromosomas más *Tit for Tat* han sido sometidos a un torneo cuyos resultados han sido los siguientes:[4]

4 La tabla recoge los resultados de los veinte primeros más *Tit for Tat*.

orden	autómata	puntos DP	3872 partidas de 192 .	Promedio p.DP/743 424	ganadas	empatadas	perdidas	Puntos Liga
1	1274	2205594	743424	2.966804945764463	232	3544	96	4008
2	975	2205576	743424	2.9667807334710745	262	3544	66	4068
3	59	2204650	743424	2.965535145488981	260	3538	74	4058
4	1303	2204552	743424	2.965403323002755	258	3544	70	4060
5	1305	2204552	743424	2.965403323002755	258	3544	70	4060
6	1306	2204552	743424	2.965403323002755	258	3544	70	4060
7	1308	2204552	743424	2.965403323002755	258	3544	70	4060
8	1309	2204552	743424	2.965403323002755	258	3544	70	4060
9	1310	2204552	743424	2.965403323002755	258	3544	70	4060
10	1311	2204552	743424	2.965403323002755	258	3544	70	4060
11	989	2204216	743424	2.9649513601928374	260	3544	68	4064
12	54	2203996	743424	2.9646554321625342	260	3536	76	4056
13	55	2203996	743424	2.9646554321625342	260	3536	76	4056
14	994	2203768	743424	2.964348743112948	256	3546	70	4058
15	1300	2203670	743424	2.964216920626722	256	3544	72	4056
16	1301	2203670	743424	2.964216920626722	256	3544	72	4056
17	1304	2203670	743424	2.964216920626722	256	3544	72	4056
18	1307	2203670	743424	2.964216920626722	256	3544	72	4056
19	988	2203638	743424	2.964173876549587	252	3546	74	4050
20	1000	2203490	743424	2.9639747976928374	258	3544	70	4060
565	tft	2157022	743424	2.9014694171831956	0	3632	240	3632

Hay que subrayar que *Tit for Tat* quedó en el puesto 565 a 0,0653 puntos (promedio) del primero. Por tanto, hay estrategias de 64 bits mejores que *Tit for Tat*. El enfrentamiento entre *Tit for Tat* y la ganadora, 1274, resulta en un empate a 576 puntos DP (promedio 3).

4.5. *Fitness* mediante un conjunto de control

Otra perspectiva es contar con un conjunto de estrategias cuyo *fitness* ha quedado probado en competencias computerizadas realizadas anteriormente. A partir de aquí he adoptado esta idea. Llamo «conjunto de control» a ese conjunto de estrategias, no todas de 64 bits. El *conjunto de control* cuenta con 40 miembros.

4.6. Evolución del conjunto de control

Describo, a continuación, como he ido generando el *conjunto de control*.

Vez 1ª: Con las estrategias de 4 bits ganadoras (ver apartados 2.1. y 2.3.) –entre las que se encuentra *Tit for Tat*– más las doscientas mejores estrategias de 16 bits, en total 209 estrategias, se ha hecho un torneo y se han seleccionado las 40 mejores estrategias. He llamado "conjunto de control-1" a estas 40 estrategias.

1. Se aplica el AG con la parametrización expuesta en el apartado 4.1.

2. Se obtuvieron 205 cromosomas de 64 bits.
3. Se llevó a cabo un torneo entre las estrategias presentes en el *conjunto de control-1* más las 205 encontradas por el AG. Cada encuentro tuvo 3×64=192 iteraciones.

Vez 2ª: Se seleccionaron las 40 mejores. 39 correspondieron a cromosomas encontrados por el AG. La estrategia de 4 bits: (1 0 0 0) se encuentra entre ellas. Se ha añadido la estrategia de 4 bits correspondiente a *Tit for Tat*: (1 0 1 0), porque no está entre las 40 mejores y porque se supone que es la mejor estrategia conocida. Nombre: "conjunto de control-2".

1. Se aplica el AG con la misma parametrización que la vez anterior.
2. Se obtuvieron 230 cromosomas de 64 bits.
3. Se llevó a cabo un torneo entre las estrategias presentes en *conjunto de control-2* más las 230 encontradas por el AG: un total de 271. Cada encuentro tuvo 3×64=192 iteraciones.

Vez 3ª: El resultado ha sido: las 271 estrategias obtuvieron igual puntos DP e igual puntos Liga porque empataron todos los encuentros: 540. Por ello, se han seleccionado 40 cromosomas de los 230 y se añadieron las dos estrategias de 4 bits: (1 0 0 0) y (1 0 1 0).

1. Se aplica el AG con la misma parametrización que la vez anterior.
2. Se obtuvieron 1.252 cromosomas de 64 bits.

4.7. Torneo entre conjuntos de control

He juntado todas las estrategias de los tres conjuntos de control en un solo conjunto de 119 elementos. Con este conjunto he llevado a cabo un torneo de todos contra todos con los siguientes resultados:

- Leyenda:
    - 4b, 16b y 64b significan respectivamente: 4 bits, 16 bits y 64 bits.
    - 64b2 y 64b3 significan respectivamente: estrategias de 64 bits del conjunto de control 2 y del conjunto de control 3.

orden	autómata	puntos DP	236 partidas de 192 jugadas	Promedio p.DP/45312	ganadas	empatadas	perdidas	Puntos Liga
1	64b2_66	126422	45312	2.79003354519774	70	162	4	302
2	64b2_65	126422	45312	2.79003354519774	70	162	4	302
3	64b2_22	126422	45312	2.79003354519774	70	162	4	302
4	64b2_50	126422	45312	2.79003354519774	70	162	4	302
5	64b2_96	126422	45312	2.79003354519774	70	162	4	302
6	64b2_6	126422	45312	2.79003354519774	70	162	4	302
7	64b2_168	126422	45312	2.79003354519774	70	162	4	302
8	64b2_67	126422	45312	2.79003354519774	70	162	4	302
9	64b2_27	126422	45312	2.79003354519774	70	162	4	302
10	64b2_166	126414	45312	2.789856991525424	70	160	6	300
11	64b2_89	126414	45312	2.789856991525424	70	160	6	300
12	64b2_84	126414	45312	2.789856991525424	70	160	6	300
13	64b2_87	126362	45312	2.788709392655367	70	162	4	302
14	64b2_25	126312	45312	2.7876059322033897	70	160	6	300
15	64b2_146	126312	45312	2.7876059322033897	70	160	6	300
16	64b2_33	126230	45312	2.785796257062147	70	162	4	302
17	64b2_55	126228	45312	2.785752118644068	70	160	6	300
18	64b2_101	126060	45312	2.782044491525424	70	160	6	300
19	64b2_121	126060	45312	2.782044491525424	70	160	6	300
20	64b2_38	126046	45312	2.7817355225988702	70	160	6	300
21	64b2_203	125976	45312	2.7801906779661016	70	160	6	300
22	64b2_178	125970	45312	2.7800582627118646	70	160	6	300
23	64b2_83	125884	45312	2.7781603107344632	70	160	6	300
24	64b2_48	125864	45312	2.777718926553672	70	162	4	302
25	64b2_183	125860	45312	2.7776306497175143	70	160	6	300
26	4b_8	125852	45312	2.7774540960451977	70	162	4	302
27	64b2_72	125820	45312	2.776747881355932	70	160	6	300
28	64b2_171	125792	45312	2.776129943502825	70	162	4	302
29	64b2_187	125734	45312	2.774849929378531	70	160	6	300
30	64b2_162	125666	45312	2.7733492231638417	70	162	4	302
31	64b2_184	125608	45312	2.772069209039548	70	162	4	302
32	64b3_196	125414	45312	2.767787782485876	70	160	6	300
33	64b2_91	125286	45312	2.7649629237288136	70	160	6	300
34	64b2_53	125282	45312	2.7648746468926553	70	162	4	302
35	64b2_204	125208	45312	2.763241525423729	70	160	6	300
36	64b2_2	125108	45312	2.7610346045197742	70	160	6	300
37	64b2_197	125076	45312	2.7603283898305087	64	166	6	294
38	64b2_44	125048	45312	2.759710451977401	70	162	4	302
39	64b2_160	124608	45312	2.75	70	162	4	302
40	64b2_161	124592	45312	2.749646892655367	70	162	4	302
44	4b_6	121272	45312	2.676377118644068	0	220	16	220

1. Entre los 40 primeros hay una estrategia de 4 bits (4b_8, el 26) y una estrategia de 64 bits del tercer conjunto de control (64b3_196, el 32). No parece exagerado afirmar que las mejores estrategias son de 64 bits y pertenecen al segundo conjunto de control.
2. Las estrategias de 16 bits quedan en los puestos finales (del 85 al 119).
3. El resto de estrategias de 4 bits se reparten así: 4b_6 (*Tit for Tat*, el 44); 4b_14, el 60; 4b_12, el 81; y 4b_16, el 82.

4.8. Torneo entre cromosomas encontrados por el AG

He reunido en un conjunto todos los cromosomas encontrados por las tres aplicaciones del AG. Y he llevado a cabo un torneo de todos contra todos incorporando a ese conjunto de estrategias de 64 bits la estrategia de *Tit for Tat*. En total 1.688 estrategias. Las veinte mejores estrategias son:

orden	autómata	puntos DP	236 partidas de 192 jugadas	Promedio p.DP/45312	ganadas	empatadas	perdidas	Puntos Liga
1	crom-694	2054894	647808	3.172072	2668	32	674	5368
2	crom-110	2004674	647808	3.094549	2864	24	486	5752
3	crom-642	1961778	647808	3.028332	2482	52	840	5016
4	crom-899	1960270	647808	3.026004	2776	32	566	5584
5	crom-166	1940390	647808	2.995316	48	3316	10	3412
6	crom-66	1940252	647808	2.995103	48	3316	10	3412
7	crom-65	1940242	647808	2.995088	48	3316	10	3412
8	crom-16	1940030	647808	2.994760	46	3320	8	3412
9	crom-807	1939964	647808	2.994658	44	3312	18	3400
10	crom-808	1939964	647808	2.994658	44	3312	18	3400
11	crom-38	1939552	647808	2.994022	40	3312	22	3392
12	crom-171	1938950	647808	2.993093	54	3316	4	3424
13	crom-146	1938946	647808	2.993087	36	3312	26	3384
14	crom-6	1938858	647808	2.992951	38	3322	14	3398
15	crom-1241	1938700	647808	2.992707	32	3314	28	3378
16	crom-161	1938546	647808	2.992469	28	3328	18	3384
17	crom-168	1938450	647808	2.992321	44	3314	16	3402
18	crom-67	1938334	647808	2.992142	44	3316	14	3404
19	crom-160	1938274	647808	2.992050	44	3314	16	3402
20	crom-178	1938260	647808	2.992028	42	3312	20	3396
98	tft	1933052	647808	2.983989	0	3346	28	3346

5. Conclusiones

Las conclusiones que creo haber alcanzado son las siguientes:

1. *Tit for Tat* es una estrategia determinista que solo atiende a una situación

precedente. Por tanto, es una estrategia de 4 bits. El torneo entre estrategias de 4 bits lo gana la estrategia de *no cooperar*, tanto en la opción A como en la B.

2. En el torneo realizado entre las 65.536 estrategias de 16 bits *Tit for Tat* no ganó. Quedó en el puesto 20378 y con una diferencia de 0,6506 puntos de promedio respecto a la primera.
3. Según el método de determinar el *fitness* de los cromosomas mediante un torneo computerizado entre generaciones (apartado 4.3.), se han encontrado estrategias de 64 bits mejores que *Tit for Tat*. En el torneo realizado entre los 1936 cromosomas más *Tit for Tat*, *Tit for Tat* quedó en el puesto 565 a 0,0653 puntos (promedio) del primero (autómata 1274).
4. El torneo entre las 1.687 estrategias encontradas por las sucesivas aplicaciones del AG, más *Tit for Tat*, da 97 estrategias de 64 bit mejores que *Tit for Tat*. Entre la primera y *Tit for Tat* hay una diferencia de 0,1880 puntos porcentuales.
5. El método de determinar el *fitnes* de cada cromosoma enfrentándolos a un conjunto de control formado por conocidas estrategias ganadoras ha mostrado su valor (apartado 4.5.), puesto que el conjunto de control evoluciona hasta conseguir algunas estrategias mejores que *Tit for Tat*.
6. En resumen, hay estrategias de 4 bits, de 16 bits y de 64 bits que son mejores que *Tit for Tat* en torneos computarizados.

6. Referencias bibliográficas

Axelrod, R. (1980): “Effective Choice in the Prisoner’s Dilemma”. The Journal of Conflict Resolution, Vol. 24, No. 1, pp. 3-25.

Axelrod, R. (1980): “More Effective Choice in the Prisoner’s Dilemma”. The Journal of Conflict Resolution, Vol. 24, No. 3, pp. 379-403.

Axelrod, R. (1984) : *The Evolution of Cooperation*. N. York, Basic Books Inc. Publisher,.

Axelrod, R. (1997): *The complexity of cooperation : agent-based models of competition and collaboration*. Princeton (N. J.), Princeton University Press.

Axelrod, R and Dion, D. (1988): “The Further Evolution of Cooperation”. Science 242, pp. 1385–1390.

Axelrod, R. and Hamilton, W. D. (1981): “The Evolution of Cooperation”. Science 211, pp. 1390–1396.

Goldberg:, D. E. (1989): *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading (MA), Addison-Wesley Pub..

Ho, Teck-Hua (1996): “Finite automata play repeated prisoner’s dilemma with information processing costs”. Journal of Economic Dynamics and Control 20, pp. 173-207.

Hofstadter, D. R. (1983): “Computer tournaments of the Prisoner’s Dilemma suggest how cooperation evolves”. Scientific American, Volume 248, Issue 5, pp. 16–26.

Holland, J. H. (1992): *Adaptation in Natural and Artificial Systems*. Cambridge (MA), The MIT Press.

Miller, J. H. (1996): “The coevolution of automata in the repeated prisoner’ s dilemma”. Journal of Economic Behavior and Organization Vol. 29, pp. 87–112.

Minsky:, M. (1967): *Computation: Finite and Infinite Machines*. Englewood Cliffs, N. J., Prentice-Hall Inc..

Mitchell, M. (1999): *An Introduction to Genetic Algorithms*. Cambridge (MA), The MIT Press.

Rubinstein, A. (1986): "Finite Automata Play the Repeated Prisoner's Dilemma". Journal of Economic Theory 39, pp. 83-96.

De Santiago, R. y García, J.A. (1994): "¿Es posible ganar a *Toma-y-Daca*?". Anales de Estudios Económicos y Empresariales, Vol. 9, pp. 159-184.

Simon, H. (1996): *The Science of the Artificial*. Cambridge, MA, The MIT Press.