Sistemas Multi-Agentes

Interacciones y Utilidad

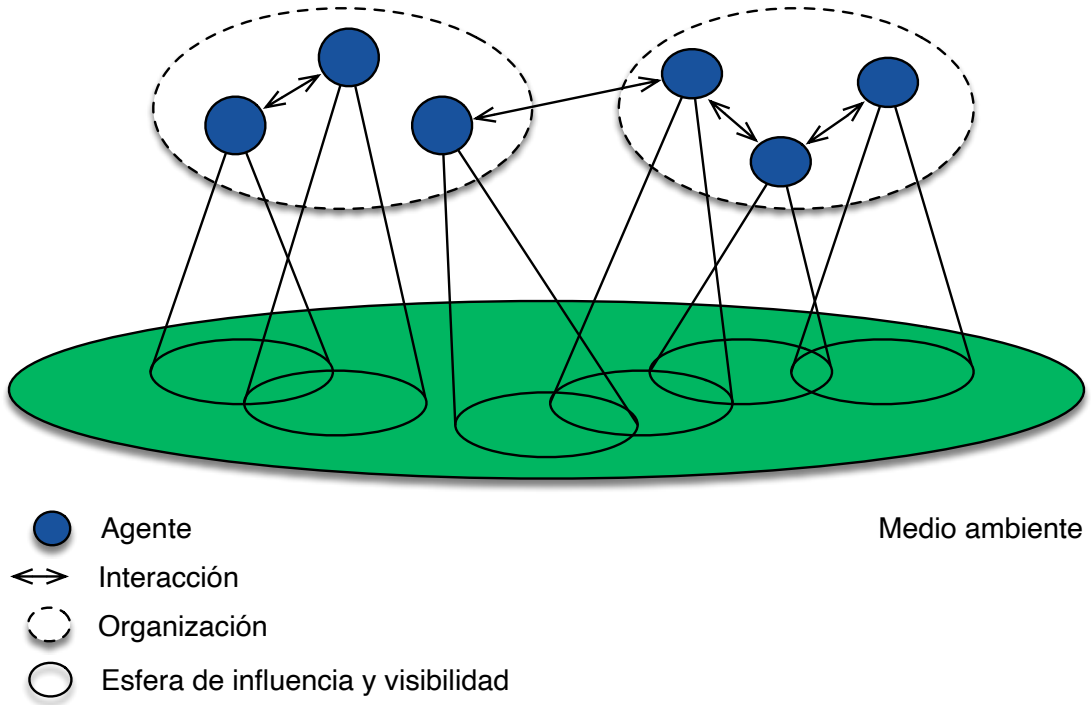
Dr. Alejandro Guerra-Hernández

Universidad Veracruzana
Instituto de Investigaciones en Inteligencia Artificial
Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,
Nuevo Xalapa, Xalapa, Ver., México 91097

aguerra@uv.mx
https://www.uv.mx/personal/aguerra/

Maestría en Inteligencia Artificial 2023

Interacciones, Organizaciones y Esferas de Influencia



- Aproximación basada en utilidades y preferencias, à la Shoham y Leyton-Brown [2].

Universidad Veracruzana

Notación

- Asumamos que tenemos dos agentes en nuestro SMA: $Ag = \{i, j\}$ con sus propios intereses o preferencias.
- El conjunto $\Omega = \{w_1, w_2, \dots\}$ representa el posible resultado de las acciones de estos agentes sobre su medio ambiente, como funciones de utilidad:

$$u_i : \Omega \rightarrow \Re$$
$$u_j : \Omega \rightarrow \Re$$

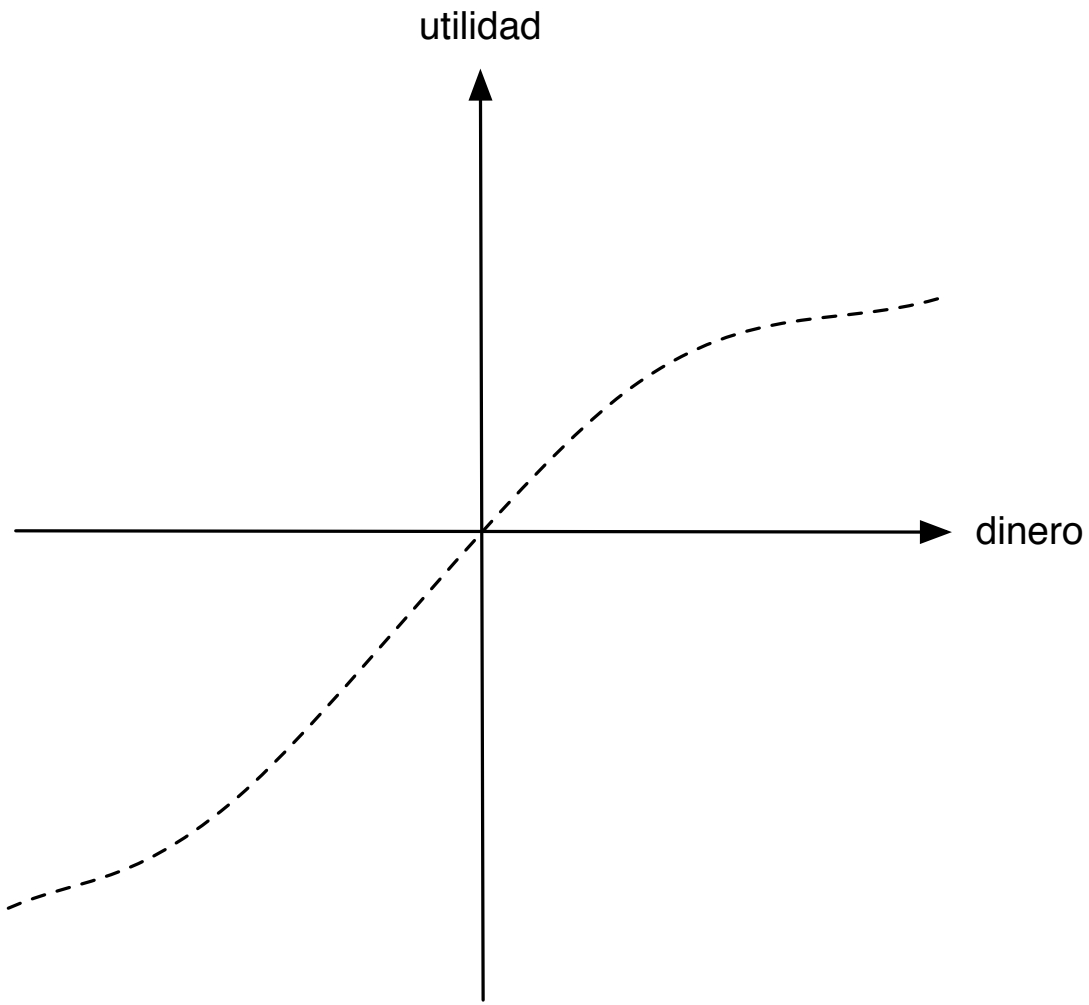
Universidad Veracruzana

Ordenes de Preferencia

- Si ω y $\omega' \in \Omega$ y $u_i(\omega) \geq u_i(\omega')$, se dice que ω es preferida por el agente i al menos tanto como ω'.
- $\omega \succeq_i \omega'$ denota que $u_i(\omega) \geq_i u_i(\omega')$; y $\omega \succ_i \omega'$ que $u_i(\omega) > u_i(\omega')$.
- La relación $\succeq_i$ es un orden sobre Ω puesto que es:
  - Reflexiva. Para todo $\omega \in \Omega$ tenemos que $\omega \succeq_i \omega$.
  - Transitiva. Si $\omega \succeq_i \omega'$ y $\omega' \succeq_i \omega''$ entonces $\omega \succeq_i \omega''$.
  - Comparable. Para todo $\omega \in \Omega$ y $\omega' \in \Omega$ tenemos que $\omega \succeq_i \omega'$ o bien $\omega' \succeq_i \omega$.
- La preferencia estricta ($\succ_i$) satisface las dos últimas propiedades, pero evidentemente no es reflexiva.

Universidad Veracruzana

Utilidad $\neq$ Dinero



Universidad Veracruzana

Simplificando el escenario

- Nos interesa modelar encuentros de agentes en el medio ambiente.
- Los dos agentes pueden elegir simultáneamente que acción llevar a cabo.
- El resultado de estas acciones está en Ω.
- Asumamos que el repertorio conductual de nuestros agentes es cooperar (C) o traicionar (D).
- Tenemos una función de transición de estados:

$$\tau = Ac_i \times Ac_j \Rightarrow \Omega$$

donde Ac_i es la acción del agente i y Ac_j la del agente j.

Universidad Veracruzana

Ejemplos

- Un ambiente que asocia una salida diferente a cada combinación de acciones de los agentes i y j:

$$\tau(D,D) = \omega_1,\ \tau(D,C) = \omega_2,\ \tau(C,D) = \omega_3,\ \tau(C,C) = \omega_4$$

  Este ambiente es sensible a las acciones de cada agente.

- Un ambiente que asocia el mismo estado a todas las combinaciones de acciones posibles:

$$\tau(D,D) = \omega_1,\ \tau(D,C) = \omega_1,\ \tau(C,D) = \omega_1,\ \tau(C,C) = \omega_1$$

  donde ninguno de los agentes tiene influencia sobre el medio ambiente.

Universidad Veracruzana

Más ejemplos

- También es posible considerar el caso donde el ambiente es sensible a las acciones de uno de los agentes:

$$\tau(D, D) = \omega_1, \ \tau(D, C) = \omega_2, \ \tau(C, D) = \omega_1, \ \tau(C, C) = \omega_2$$

  donde el ambiente es controlado por por el agente j.
- Si el agente j decide traicionar (D), la salida será ω_1 y si decide cooperar (C) será ω_2.

Universidad Veracruzana

Utilidades, Ambiente y Preferencias

- Supongamos que los agentes tienen las funciones de utilidad siguientes:

$$u_i(\omega_1) = 1 \quad u_i(\omega_2) = 1 \quad u_i(\omega_3) = 4 \quad u_i(\omega_4) = 4$$
$$u_j(\omega_1) = 1 \quad u_j(\omega_2) = 4 \quad u_j(\omega_3) = 1 \quad u_j(\omega_4) = 4$$

- Y abusando un poco de la notación:

$$u_i(D, D) = 1 \quad u_i(D, C) = 1 \quad u_i(C, D) = 4 \quad u_i(C, C) = 4$$
$$u_j(D, D) = 1 \quad u_j(D, C) = 4 \quad u_j(C, D) = 1 \quad u_j(C, C) = 4$$

- Tenemos que, para el agente i:

$$C, C \succeq_i C, D \succ_i D, C \succeq_i D, D$$

Universidad Veracruzana

¿Cooperar o Traicionar?

- La respuesta no tiene ambigüedad, el agente prefiere las situaciones donde coopera a aquellas donde traiciona.
- La opción del agente i es cooperar independientemente de lo que haga el agente j.
- De igual manera, el agente j prefiere las situaciones donde coopera.
- Observen que en este caso, ninguno de los agentes debe preocuparse por lo que hace el otro, la acción que eligen no depende de la acción del otro.
- La acción conjunta será C, C: ambos agente cooperarán.

Universidad Veracruzana

Otras preferencias

- Ahora supongamos que, para el mismo medio ambiente, las funciones de utilidad son las siguientes:

$$u_i(D,D) = 4 \quad u_i(D,C) = 4 \quad u_i(C,D) = 1 \quad u_i(C,C) = 1$$
$$u_j(D,D) = 4 \quad u_j(D,C) = 1 \quad u_j(C,D) = 4 \quad u_j(C,C) = 1$$

- Las preferencias del agente i son como sigue:

$$D,D \succeq_i D,C \succ_i C,D \succeq_i C,C$$

Universidad Veracruzana

¿Qué hacer?

- En este escenario, el agente *i* no puede hacer nada mejor que traicionar.
- El agente prefiere las salidas donde traiciona a todas aquellas donde coopera.
- De igual manera, el agente *j* no puede hacer nada que traicionar.
- Nuevamente los agentes no tienen porque dedicar recursos a elaborar una estrategia basada en qué hará el otro agente.
- En la mayoría de los SMA los encuentros no son tan claros como en estos dos ejemplos.

Universidad Veracruzana

Matriz de Pago

	i traiciona	i coopera
j traiciona	4 4	1 4
j coopera	4 1	1 1

Dominancia

- Asumamos dos subconjuntos de Ω denotados por Ω_1 y Ω_2.
- Se dice que Ω_1 domina a Ω_2 para el agente i si cada estado de Ω_1 es preferido por i sobre cualquier estado de Ω_2. Por ejemplo:
  - $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$
  - $\omega_1 \succ_i \omega_2 \succ_i \omega_3 \succ_i \omega_4$
  - $\Omega_1 = \{\omega_1, \omega_2\}$
  - $\Omega_2 = \{\omega_3, \omega_4\}$
- Es evidente que Ω_1 domina fuertemente a Ω_2. Formalmente:

$$\forall \omega_1 \in \Omega_1, \forall \omega_2 \in \Omega_2 \ \omega_1 \succ_i \omega_2$$

Universidad Veracruzana

Estrategias (teoría de juegos)

- Nos referiremos a las acciones de los agentes (miembros de Ac) como estrategias.
- Dada una estrategia particular s para un agente i en un escenario de interacción SMA, habrá un número posible de estados resultantes.
- Denotaremos s^* a los estados que resulten de la estrategia s.
- En los ejemplos de la sección anterior, desde el punto de vista del agente i tenemos que $C^* = \{\omega_3, \omega_4\}$ y $D^* = \{\omega_1, \omega_2\}$.
- El concepto de dominancia aplica a las estrategias.

Universidad Veracruzana

Equilibrio de Nash

- Decimos que dos estrategias están en equilibrio si:
  1. Bajo el supuesto de que el agente i jugará con la estrategia s_1, el agente j no puede hacer nada mejor que jugar con la estrategia s_2; y
  2. Bajo el supuesto de que el agente j jugará con la estrategia s_2, el agente i no puede hacer nada mejor que jugar con la estrategia s_1.

Universidad Veracruzana

Buenas y Malas Nuevas

- Ningún agente tiene incentivo alguno para salir del equilibrio de Nash.
- Desafortunadamente hay dos resultados de la teoría de juegos con respecto al equilibrio:
  1. No todo escenario de interacción tiene un equilibrio de Nash; y
  2. Algunos escenarios de interacción tienen más de un equilibrio de Nash.

Universidad Veracruzana

Encuentros estrictamente competitivos

- Supongamos que tenemos un estado $\omega \in \Omega$ que es preferido por el agente i con respecto a un estado ω', si y sólo si ω' es preferido por el agente j con respecto a ω. Formalmente:

$$\omega \succ_i \omega' \text{ si y sólo si } \omega' \succ_j \omega$$

- Las preferencias de los agentes son diametralmente opuestas, de forma que un agente puede incrementar su utilidad sólo a costa del otro agente.

Universidad Veracruzana

Juegos de Suma Cero

- Son aquellos en los cuales para cualquier estado alcanzado, las utilidades de los dos agentes suman cero. Formalmente:

$$\forall \omega \in \Omega \; u_i(\omega) + u_j(\omega) = 0$$

- Todo encuentro de suma cero es estrictamente competitivo.
- Son el caso más viciado de interacción, donde no hay posibilidad de comportamiento cooperativo.
- Si un agente permite que el otro agente tenga utilidad positiva, el tendrá utilidad negativa y estará peor que antes de llevar a cabo la interacción.

Universidad Veracruzana

Descripción

- Dos hombres son acusados de un crimen colectivo y se les mantiene en celdas separadas.
- No tienen forma de comunicarse entre si o de llegar a alguna forma de acuerdo.
- A ambos se les dice que:
  1. Si uno de ellos confiesa el crimen y el otro no lo hace, el confeso será liberado y al otro se le darán tres años de cárcel.
  2. Si ambos confiesan el crimen estarán dos años en la cárcel.
- Ambos prisioneros saben que si ninguno confiesa el crimen, entonces cada uno será encarcelado un año.

Universidad Veracruzana

Una posible matriz de pago

	i traiciona	i coopera
j traiciona	2 2	0 5
j coopera	5 0	3 3

- Las utilidades son:

$$u_i(D,D) = 2 \quad u_i(D,C) = 5 \quad u_i(C,D) = 0 \quad u_i(C,C) = 3$$
$$u_j(D,D) = 2 \quad u_j(D,C) = 0 \quad u_j(C,D) = 5 \quad u_j(C,C) = 3$$

- y sus preferencias son:

$$D,C \succ_i C,C \succ_i D,D \succ_i C,D,$$
$$C,D \succ_j C,C \succ_j D,D \succ_j D,C.$$

¿Qué hacer?

- *i* coopera. Si *j* coopera, obtendrán ambos agentes un pago igual a tres. Pero si *j* traiciona, el agente *i* obtendría un pago de cero, de forma que el mejor pago que *i* puede obtener cooperando es cero.
- *i* traiciona. Sii *j* coopera, *i* obtendrá un pago igual a cinco, mientras que si *j* traiciona, obtendría un pago igual a dos. De forma que el mejor pago que puedo garantizar es dos.
- El agente *i* preferirá garantizar un pago igual a dos, que obtener un pago igual a cero y por lo tanto, traicionará al agente *j*.

Universidad Veracruzana

Contrabando y Traición (La Sombra del Futuro)

- Consideremos que el juego se repite indefinidamente, ronda tras ronda. Las razones para traicionar se desvanecen por las siguiente dos razones:
  - Si i traiciona ahora, el oponente j puede castigarle traicionando más tarde. El castigo no es posible en la definición no iterada del dilema.
  - Si i tantea cooperar y recibe el pago del tonto en el juego en la primera ronda, entonces debido a que se juega indefinidamente, esta perdida de utilidad puede ser amortizada en los juegos futuros. En la perspectiva de que el juego es infinito, la perdida de una sola unidad de utilidad representa un pequeño porcentaje de la utilidad total ganada.
- ¿Qué pasa con 100 rondas (finito)?

Universidad Veracruzana

Torneo de Axelrod

- Axelrod [1] es un politólogo interesado en como la cooperación puede emerger en una sociedad de agentes individualistas.
- En 1980 organizó un torneo público en el cual politólogos, psicólogos, economistas y teóricos del juego fueron invitados a enviar un programa que jugará el dilema del prisionero iterado.
- Cada programa de computadora tenía disponible las opciones previas de su contrincante y simplemente seleccionada C o D basado en esa información.
- Cada programa jugaba contra los demás programas 5 juegos de 200 rondas.

Universidad Veracruzana

Estrategias

Todo-D. La estrategia racional es siempre traicionar.

Aleatoria. Selecciona C o D aleatoriamente, con la misma probabilidad.

Ojo por ojo. Conocida también como tit for tat:

1. En la primera ronda el agente coopera;
2. en las rondas $t > 1$ hace lo que su oponente hizo en la ronda $t - 1$.

Probador. En la primer ronda traiciona. Si el oponente contesta traicionado, en las rondas subsecuentes el agente juega ojo por ojo. Si el otro agente no traiciona, coopera dos rondas y luego traicionando una vez más.

Joss. Se comporta siguiendo una estrategia ojo por ojo, sólo que traiciona el 10 % del tiempo, en lugar de cooperar.

Universidad Veracruzana

Moralejas

- No seas envidioso. En el dilema del prisionero no es necesario derrotar al oponente para jugar bien.
- No seas el primero en traicionar. Aunque comenzar a jugar cooperando es riesgoso, la pérdida de traición en la primera ronda es pequeña comparada con las posibilidades de cooperar.
- Se reciproco en la cooperación y la traición. La estrategia ojo por ojo representa un balance entre castigar y ser perdonado: la combinación de castigar la traición y premiar la cooperación parece motivar la cooperación.
- No seas demasiado listo. La estrategia ojo por ojo es la más simple de todas las estrategias y programas presentados.

Universidad Veracruzana

Exito de Ojo por Ojo

- Los programas más complejos intentan crear un modelo del comportamiento del otro agente, ignorando el hecho de que el otro agente está observando al agente original – carecen de un modelo de aprendizaje recíproco, que es lo que sucede en realidad.
- Los programas más complejos sobregeneralizan cuando encuentran el defecto del oponente y no permiten que la cooperación sea aún posible en el futuro – no perdonan.
- Muchos de los programas complejos exhibían un comportamiento demasiado complicado para ser entendido por el otro agente – el oponente creía que su comportamiento era aleatorio.

Universidad Veracruzana

Matriz de pago

- Representamos la matriz de pago con creencias:

```
/* Matriz de Pago
 * formato: u(Acción_ag1,Acción_ag2,Utilidad_ag1,Utilidad_ag2)
 * Usada por todos los agentes jugadores (Conociminto común)
*/

u(c,d,0,5). // si traiciono al que coopera tomo 5
u(d,c,5,0). // si me traicionan cooperando me voy con 0
u(c,c,3,3). // recompensa por cooperación mutua = 3
u(d,d,2,2). // castigo por traición mutua = 2
```

Universidad Veracruzana

Jugador I

- El jugador genérico es como sigue:

```
1  /*** creencias iniciales ***/
2
3  mi_utilidad(0).
4
5  // Matriz de pago conocida por todos los agentes
6  {include("matrizPago.asl")}
7
8  /*** metas iniciales ***/
9
10 !inicio.
11
12 /*** planes ***/
13
14 // Me presento con el arbitro
15 +!inicio <-
16   .my_name(Yo);
17   .send(arbitro, tell, jugador(Yo)).
18
19 // El resultado del encuentro ha sido anunicado por el
```

Universidad Veracruzana

Jugador II

```
20  // arbitro: actualizar score y registrar resultados por
21  // si mi estrategia lo requiere
22  @u[atomic]
23  +u(Paso,Utilidad)[source(arbitro)] : arrestado(Paso,Oponente) <-
24    -u(_,_);
25    +u(Paso,Utilidad);
26    ?mi_utilidad(U);
27    UtilidadNueva = U+Utilidad;
28    -+mi_utilidad(UtilidadNueva); // actualizar utilidad total
29    -arrestado(Paso,Oponente)[source(arbitro)]; // olvidar mi arresto
30    -u(Paso,Utilidad)[source(arbitro)]; // olvidar utilidades, no las
      ↪ necesito
31    ?u(_,AccOponente,Utilidad,_); // Si obtuve Utilidad el otro actuó
      ↪ AccOponente
32    !registrar(Paso,Oponente,AccOponente); // algunas estrategias lo requiren
33    dpi.plot(Paso,UtilidadNueva);   // acción interna para graficar
      ↪ utilidades
34    .print("Obtuve ",Utilidad," en el paso ",Paso,". Mi total es ahora: ",
35           UtilidadNueva).
```

Universidad Veracruzana

El arbitro I

- El arbitro del torneo es como sigue:

```
1  /* creencias iniciales */
2
3  paso(0).
4  rondas(300).
5
6  // Creencias comunes sobre la utilidad del dilema del prisionero
7  {include("matrizPago.asl")}
8
9  // Regla para seleccionar dos jugadores al azar
10 seleccionar(Jugador1,Jugador2) :-
11   .count(jugador(_),N)  &
12   dos_rands(R1,R2,N) &
13   .findall(J,jugador(J),ListaJugadores) &
14   .nth(R1,ListaJugadores,Jugador1) & .nth(R2,ListaJugadores,Jugador2).
15
16 // Regla para obtener dos números aleatorios "diferentes" menores a N
17 // dpi.random es una acción interna con backtracking
18 dos_rands(R1,R2,N) :-
19   dpi.random(R1,N) & dpi.random(R2,N) & R1 \== R2.
```

Universidad Veracruzana

El arbitro II

```
20
21  /* planes */
22
23  // Tan pronto como sepa de 2 jugadores puedo arrestarlos
24  +jugador(_) :  .count(jugador(_),NoAgs) & NoAgs >= 2 <-
25    !!arrestar.
26
27  // Tengo la meta de arrestar a dos jugadores
28  @arrestar[atomic]
29  +!arrestar : paso(Paso) & rondas(Ronda) & Paso <= Ronda <-
30    ?seleccionar(Jugador1,Jugador2); // Selecciona dos jugadores al azar
31    .print("Arrestando a ",Jugador1," y ",Jugador2," (paso ",Paso,")");
32    .send(Jugador1,tell,arrestado(Paso,Jugador2));
33    .send(Jugador2,tell,arrestado(Paso,Jugador1));
34    -+paso(Paso+1).
35
36  // He terminado el número de rondas establecidas: no hago nada.
37  +!arrestar.
38
39  // Qué jugó el segundo de los jugadores?
40  // De forma que pueda calcular las utilidades y comunicarlas
```

Universidad Veracruzana

El arbitro III

```
41  @jugar[atomic]
42  +jugar(Paso,AccAg1)[source(Jugador1)]
43    :  jugar(Paso,AccAg2)[source(Jugador2)] & Jugador1 \== Jugador2 <-
44    ?u(AccAg1,AccAg2,UAg1,UAg2);
45    .print("Utilidades en el paso ",Paso,": ",UAg1," ",UAg2);
46    .send(Jugador1,tell,u(Paso,UAg1));
47    .send(Jugador2,tell,u(Paso,UAg2));
48    .abolish(jugar(Paso,_)[source(_)]);
49    !!arrestar.
50
51  +jugar(Paso,_). // si solo conozco una jugada no hago nada.
```

Universidad Veracruzana

Todo D

- Don mala gente es como sigue:

```
1  /*** Soy un jugador.. ***/
2
3  { include("jugador.asl") }
4
5  /*** Y mi estrategia es ... ***/
6
7  // Siempre traicionar
8  +arrestado(Paso,Oponente)[source(arbitro)] <-
9    .send(arbitro, tell, jugar(Paso,d));
10   .print("Traiciono como siempre, a ",Oponente," en el paso ",Paso).
11
12 // No necesito registros: no hace nada
13 +!registrar(Paso,Oponente,AccOponente).
14
```

Universidad Veracruzana

Todo C

- Don buena gente es como sigue:

```
1  /*** Soy un jugador...  ***/
2
3  { include("jugador.asl") }
4
5  /*** Y mi estrategia es ... ***/
6
7  // Siempre cooperar
8  +arrestado(Paso,Oponente)[source(arbitro)] <-
9    .send(arbitro, tell, jugar(Paso,c));
10   .print("Coopero como simpre con ",Oponente," en el paso ",Paso).
11
12 // No necesito registros nuevoss
13 +!registrar(Paso,Oponente,AccOponente).
14
```

Universidad Veracruzana

Ojo por Ojo I

- Don justo es como sigue:

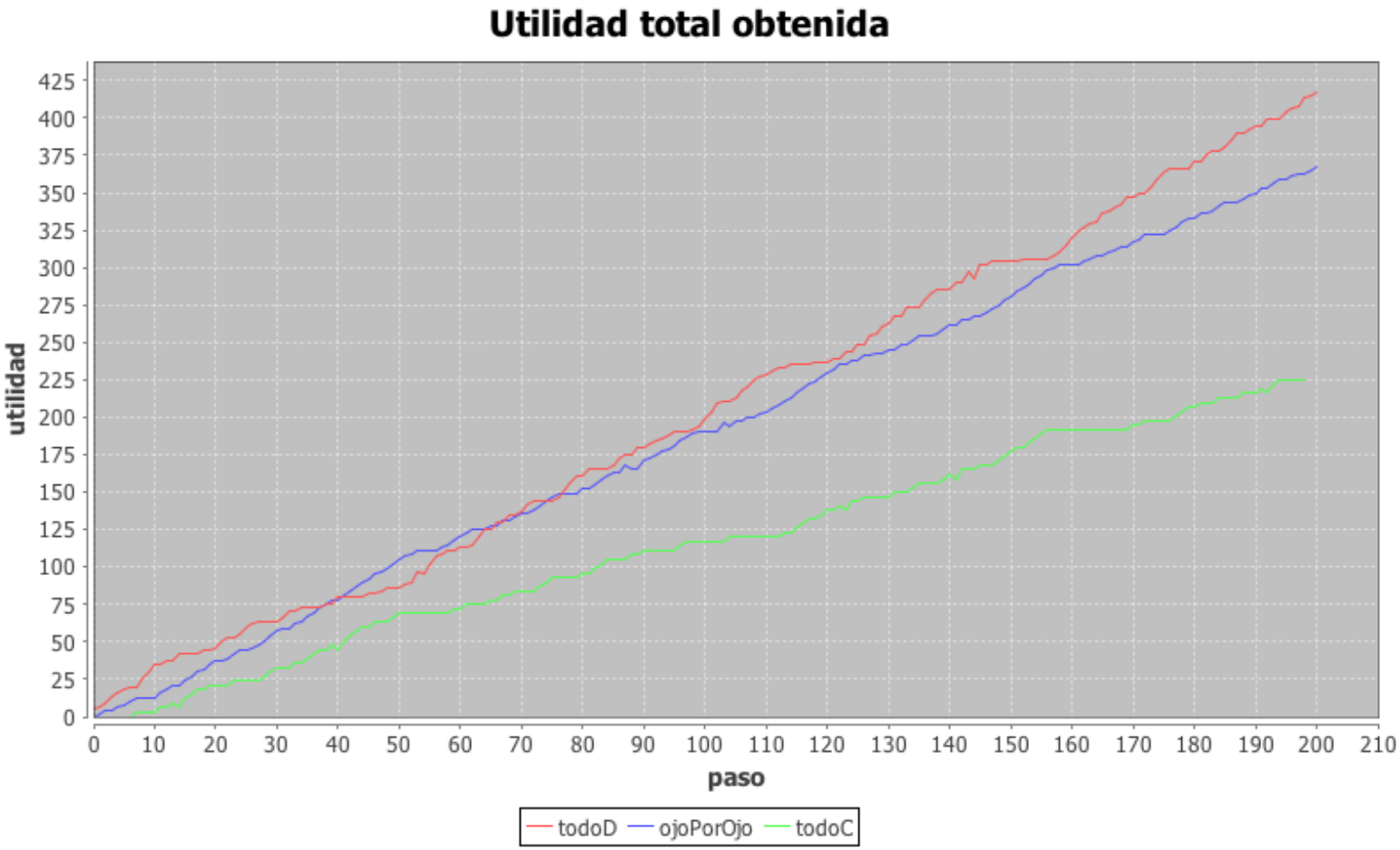
```
1  /*** Soy un jugador... ***/
2
3  {include("jugador.asl")}
4
5  /*** Que necesita acordarse del juego del oponente ... ***/
6
7  @registrar[atomic]
8  +!registrar(Paso,Oponente,AccOponente) <-
9    -ultimo_mov(Oponente,_); // NB: cannot use -+ here!
10   +ultimo_mov(Oponente,AccOponente).
11
12 /*** Mi estrategia es... ***/
13
14 // Ojo por ojo y diente por diente
15 // (1) Juego lo mismo que  jugo el oponente la última vez
16 +arrestado(Paso,Oponente)[source(arbitro)]
17   :  ultimo_mov(Oponente,UltimaJugadaOponente)
18   <-
19   .send(arbitro, tell, jugar(Paso,UltimaJugadaOponente));
```

Ojo por Ojo II

```
20      .print("Jugué ",UltimaJugadaOponente," contra ",Oponente,
21             " en el paso ",Paso).
22
23    // (2) Soy cortés la primera vez que juego (coopero)
24    +arrestado(Paso,Oponente)[source(arbitro)] <-
25      .send(arbitro, tell, jugar(Paso,c));
26      .print("Cooperé contra ",Oponente," en el paso ",Paso).
27
```

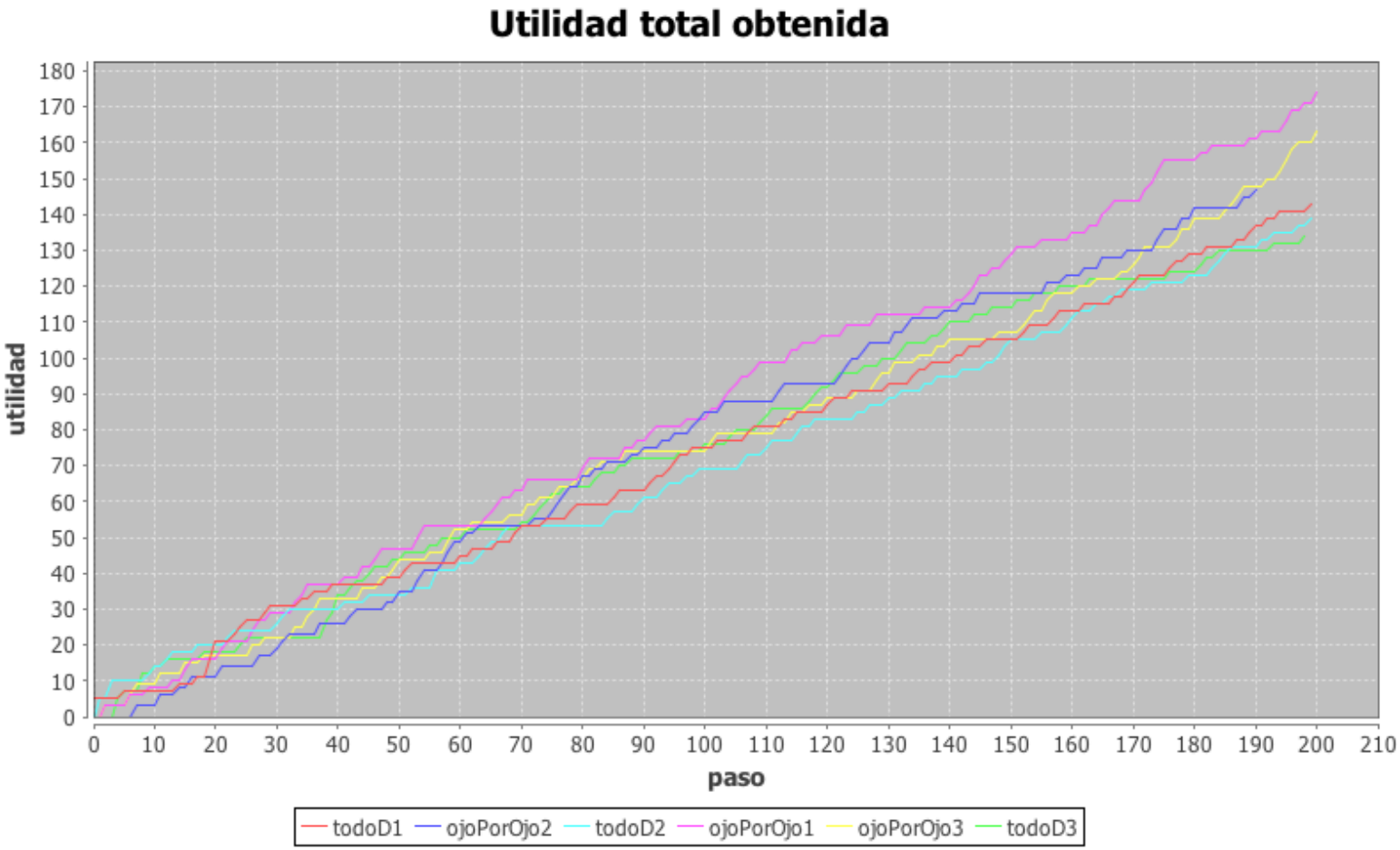
Universidad Veracruzana

Traicionar paga, si los demás se dejan



Universidad Veracruzana

Si no, no siempre paga



Universidad Veracruzana

Referencias I

[1] R Axelrod. "More effective choice in the prisoner's dilemma". En: *Journal of Conflict Resolution* 24.3 (1980), págs. 379-403.

[2] Y Shoham y K Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

Universidad Veracruzana