

Problema 2 de 4

SID 2122q2

Marzo/Abril 2022

El dilema del prisionero¹ es un juego no cooperativo y de suma no nula² que trata de modelar un ejemplo paradigmático de cómo dos agentes racionales pueden decidir no cooperar, *aunque esta decisión vaya en contra de los intereses de ambos*. El dilema del prisionero generalmente se enuncia de la siguiente manera:

Dos sospechosos de cometer un crimen (agentes α y β) son arrestados y están siendo interrogados por separado sin poder comunicarse entre si. Para conseguir una confesión, la policía les ofrece un trato. Si ambos acusan al otro del crimen, ambos serán condenados a 3 años. Si ninguno de los dos habla, ambos serán condenados a 1 año. Sin embargo, si α acusa a β y β no habla, α saldrá en libertad y β cumplirá 6 años de condena. Igualmente, si β decide acusar a α y α calla, α cumplirá 6 años y β quedará libre.

La matriz de penalizaciones $P(\text{agent})$ se resumiría en la siguiente tabla (no acusar/callar \rightarrow *cooperate*, acusar/traicionar \rightarrow *defect*):

	β C(operates)	β D(efects)
α C(operates)	$P(\alpha) = P(\beta) = 1$	$P(\alpha) = 6, P(\beta) = 0$
α D(efects)	$P(\alpha) = 0, P(\beta) = 6$	$P(\alpha) = P(\beta) = 3$

En la forma básica del juego, los dos agentes eligen C o D simultáneamente, sin comunicarse entre ellos. En la forma general (*iterated prisoner's dilemma*) el juego simultáneo se repite durante un número arbitrario de turnos, acumulando las penalizaciones.

En este enunciado usaremos una versión modificada (*alternating prisoner's dilemma* o *turn-taking prisoner's dilemma*) en la cual los agentes eligen C o D de manera alternativa, acumulando las penalizaciones por pares de decisiones. Por ejemplo, si α y β juegan alternativamente C, D, D, C se entiende que en el primer turno se ha jugado $\alpha \rightarrow C$ y $\beta \rightarrow D$ (penalización de 6 para α y 0 para β) y en el segundo se ha jugado $\alpha \rightarrow D$ y $\beta \rightarrow C$ (penalización de 0 para α y 6 para β), resultando en una penalización acumulada de 6 para cada agente.

¹Rapoport, Anatol, Albert M. Chammah, and Carol J. Orwant. Prisoner's dilemma: A study in conflict and cooperation. Vol. 165. University of Michigan press, 1965.

²Lo que gana un agente puede ser mayor o menor que lo que pierde el otro.

1 Enunciado

Usando la plataforma de JADE y las librerías de JADE y BDI4JADE, implementad un agente que juegue al dilema del prisionero con una estrategia muy básica. Este agente debe cumplir los siguientes requerimientos:

- Debe tener como mínimo una *capability*.
- Recibe las penalizaciones para el agente como argumentos en su creación, por este orden: CC , CD , DC y DD , donde XY se interpreta como la penalización para el agente cuando éste juega X y su adversario juega Y , independientemente de quién haya hecho su jugada antes. Siguiendo el ejemplo anterior, estos valores serían $CC = 1$, $CD = 6$, $DC = 0$, $DD = 3$.
- Ha de tener un estado interno que permita guardar:
 - La tabla de penalizaciones para el agente (CC , CD , DC y DD).
 - Qué agentes hay disponibles para jugar.
 - Un histórico de jugadas que permita identificar las partidas en juego: con qué agente y en qué conversación se juega cada partida y qué ha jugado cada agente.
 - La penalización acumulada.
- Debe perseguir cuatro objetivos, que a su vez pueden contener subobjetivos:
 - Registrarse en el DirectoryFacilitator con *serviceType* == 'player'.
 - Encontrar agentes en el DirectoryFacilitator con los que jugar (*serviceType* == 'player').
 - Proponer jugada:
 1. Escoger un agente β al azar entre los disponibles para jugar con los que no haya una partida en marcha.
 2. Decidir una jugada según el siguiente criterio:
 - * Escoger C (cooperate) si $\min(CC, CD) \leq \min(DC, DD)$.
 - * Escoger D (defect) si $\min(CC, CD) > \min(DC, DD)$.
 3. Enviar al agente β un mensaje INFORM, con ontología 'play' y con la jugada decidida (*content* == 'C' o *content* == 'D').
 4. Registrar la jugada en el estado interno del agente.
 - Esperar jugada:
 1. Recibir mensajes de tipo INFORM, con ontología 'play', y con una jugada C o D como contenido.
 2. Registrar la jugada en el estado interno del agente.
 3. Decidir la siguiente jugada en base al siguiente criterio:

- * Si la jugada recibida es C : escoger C si $CC \leq DC$, en caso contrario, escoger D .
 - * Si la jugada recibida es D : escoger C si $CD \leq DD$, en caso contrario, escoger D .
 - 4. Enviar al agente α un mensaje de respuesta INFORM, con ontología ‘play’ y con la jugada decidida ($content == ‘C’$ o $content == ‘D’$).
 - 5. Registrar la jugada en el estado interno del agente.
- El agente debería iniciar una nueva partida siempre que aparezca un agente nuevo con el cual no se estaba jugando aún. Para mantener los objetivos relevantes activos, podéis usar (justificadamente) el mecanismo que queráis para ello, por ejemplo:
 - Añadir objetivos en tiempo de ejecución, ya sea desde los *plan bodies* o de manera externa, e.g. desde un *TickerBehaviour* definido en el `init()` del agente.
 - Definir *plan bodies* que no acaben.
 - Usar *MessageTemplates* como objetivo de los planes.
 - Suscribirse al DF en lugar de hacer una búsqueda síncrona en bucle para recibir notificaciones push cuando aparezca un agente nuevo. Esto se puede hacer enviando al DF un mensaje SUBSCRIBE, creado usando el método *createSubscriptionMessage* de `DFService`³.

Ha de ser posible iniciar en la plataforma de JADE un número arbitrario de instancias de agente basadas en el código de vuestro agente. Con independencia de la cantidad de agentes (y de si son todos vuestros o no), cada uno de ellos ha de poder funcionar de manera autónoma en la plataforma. Vigilad especialmente que no se sature la plataforma debido a bucles de behaviours o bucles dentro de behaviours.

2 Entrega

En la entrega tenéis que incluir:

- Un documento breve (.txt o .pdf) que explique:
 - Las decisiones de diseño que habéis seguido para implementar el agente.
 - Cómo os habéis repartido las tareas entre los autores de la entrega.
- El código fuente del agente.

³<https://jade.tilab.com/doc/api/jade/domain/DFService.html>

3 Evaluación

La evaluación se dividirá en los siguientes aspectos:

- Las instancias basadas en este agente funcionan correctamente (cumple con los requerimientos) y de manera estable. (5/10 puntos)
- El diseño de objetivos, planes y plan bodies, así como de (si existen) behaviours definidos fuera del ciclo BDI, se corresponde con los requerimientos del agente. (2/10 puntos)
- El *DirectoryFacilitator* se utiliza correctamente para el registro y descubrimiento de los agentes. (1/10 puntos)
- El estado interno está correctamente diseñado y se encapsula en beliefs. (2/10 puntos)
- *Extra*: La estrategia de selección de jugada es más sofisticada que la que se propone y proporciona mejores resultados a la larga. Podéis investigar las diferentes posibilidades investigando sobre estrategias ya existentes para el problema del dilema del prisionero iterativo (*iterated prisoner's dilemma*) o por turnos (*alternating/turn-taking prisoner's dilemma*). (2/10 puntos)

La entrega y corrección del problema se hará en grupos de como máximo 3 personas y contará un 1/4 de la nota de problemas. Deberéis entregar vuestra solución antes del final del viernes 22 de abril en el espacio que se habilitará en el Racó.