Guía para el diseño y análisis de agentes en un juego repetido

A continuación se propone una **guía** para que estudiantes diseñen y analicen **agentes (estrategias)** que participen en este juego repetido de esfuerzo/ocio/calificación, inspirándose en la lógica de los torneos tipo Axelrod pero **adaptados a este contexto**. La idea es que cada estudiante programe o describa un **agente** que tome decisiones de esfuerzo en sucesivos trabajos, conociendo u observando (según el caso) las acciones pasadas de sus compañeros y la calificación obtenida. Luego, esos agentes se enfrentan en **diversos escenarios** (diferentes funciones de calificación, estructuras de preferencia, horizontes temporales, etc.) y se comparan sus desempeños.

---

1. Estructura general del problema

1. **Juego repetido**:
    - Hay T rondas (o trabajos).
    - En cada ronda, el equipo de E estudiantes (agentes) simultáneamente decide cuántas horas $h_{e,t}$ dedicar.
    - Se obtiene una calificación $G_t = f(\sum_{e=1}^E h_{e,t})$.
    - Cada agente obtiene una utilidad que combina dicha calificación (compartida) y su ocio (horas no invertidas).
2. **Variabilidad de escenarios**:
    - Funciones f con distinta forma (lineal, saturación rápida, concavidad, etc.).
    - Preferencias de ocio distintas (lineales, decrecientes, logarítmicas, etc.).
    - Jugadores con distintos pesos de calificación (α_e) o productividades (ω_e).
    - Diferentes horizontes T (corto vs. largo plazo).
    - Distintos tamaños de equipo E.

PROF

3. **Torneo o simulación masiva**:
    - Se crean varios tipos de agentes (estrategias).
    - Cada agente se empareja aleatoriamente con grupos distintos a lo largo de múltiples "partidas" o "simulaciones".
    - Se registran los resultados (calificaciones finales, horas de ocio, utilidad global, etc.) para comparar qué estrategia fue más exitosa en promedio.

---

2. Diseño de agentes y estrategias

La pregunta central es: **¿Cómo decide cada agente cuánto esfuerzo poner en cada ronda, dados los resultados pasados y las características del juego?** A continuación se ofrecen **varios enfoques**:

2.1. Estrategias fijas o de umbral

1. **Estrategia constante**:
    - El agente siempre dedica h^* horas, sin importar el comportamiento de los demás.
    - Útil como baseline para comparar con otras estrategias.
2. **Estrategia de umbral (threshold)**:
    - El agente elige horas $h_{e,t}$ si observa (o anticipa) que el total de horas de los demás, $\sum_{i\neq e}h_{i,t}$, sea menor/ mayor que cierto umbral.
    - "Si mis compañeros no se esfuerzan, mejor no poner mis horas" (comportamiento punitivo) o "aporto más solo si los demás ya están aportando".
3. **Estrategia escalonada**:
    - Comienza con un cierto esfuerzo alto. Si detecta free-riders, lo reduce gradualmente a modo de "castigo".
    - Variación inspirada en la "titulación" (tit-for-tat) del Dilema del Prisionero repetido.

> **Ventaja**: Muy sencillas de programar y rápidas de entender.
> **Desventaja**: Pueden no adaptarse bien a escenarios muy cambiantes (por ejemplo, cuando la productividad ω_e difiere mucho entre agentes).

2.2. Estrategias adaptativas con aprendizaje

1. **Aprendizaje por refuerzo (Q-learning o similar)**:
    - El agente trata de **maximizar** su utilidad esperada aprendiendo qué acción (nivel de horas) funciona mejor en distintas "circunstancias" (historial de calificaciones, acciones de otros).
    - Con el tiempo, el agente puede "descubrir" qué niveles de esfuerzo le convienen dada la dinámica del grupo.
2. **Modelos evolutivos**:
    - Se generan muchas copias de cada tipo de estrategia; las más exitosas se replican, las menos exitosas desaparecen.
    - Se observa qué estrategias emergen con mayor frecuencia y "dominan" a largo plazo.

PROF

> **Ventaja**: Pueden ajustarse mejor a entornos complejos y cambiar su comportamiento si la situación lo requiere.
> **Desventaja**: Requieren más tiempo de entrenamiento y más parámetros de diseño (tasa de aprendizaje, estados, recompensas inmediatas vs. futuras, etc.).

2.3. Estrategias basadas en reputación y señales

1. **Uso de la historia colectiva**:
    - El agente guarda un registro de cuántas horas ha aportado cada compañero en rondas anteriores.
    - Ajusta su esfuerzo: premia a quienes colaboran "compensando su esfuerzo" y castiga a quienes free-ride "rebajando el propio".

2. **Sistemas de puntuación interna**:
    - Cada agente asigna "puntos de reputación" a los demás, en función de su contribución.
    - Un compañero con buena reputación (alto esfuerzo pasado) recibe cooperación; uno con mala reputación, no.

> **Ventaja**: Explícito para modelar la cooperación a largo plazo y la credibilidad de castigos.
> **Desventaja**: Puede volverse complejo con muchos jugadores y exige que se comparta o se conozca esa reputación.

---

3. Pasos para realizar la simulación o torneo

A grandes rasgos, la mecánica para un "torneo" computacional podría ser:

1. **Definir los parámetros** de la simulación:
    - Número de rondas T.
    - Tamaño de los equipos E.
    - Función de calificación f.
    - Forma de la utilidad del ocio (lineal, $\sqrt{\cdot}$, $\log(\cdot)$, etc.).
    - Valoraciones de la calificación (α_e) y productividades (ω_e) para cada agente (si aplican).
2. **Generar un conjunto de agentes** (estrategias) a comparar:
    - Por ejemplo: "Constante baja", "Constante media", "Condicional al esfuerzo de los demás", "Aprendiz Q-learning", "Basado en reputación", etc.
3. **Asignar equipos de forma aleatoria** repetidas veces:
    - En cada "partida", se sortean E agentes de entre los disponibles.
    - Se ejecuta el juego durante T rondas con esos agentes en el equipo.
    - Se registran las utilidades finales de cada agente.

PROF

4. **Repetir con múltiples instancias**:
    - Para obtener resultados robustos, correr varias simulaciones con distintas semillas aleatorias, distintos valores de α_e, distintos ω_e, etc.
5. **Analizar los resultados**:
    - Promedio de la utilidad de cada agente (o de cada tipo de estrategia).
    - Distribución de esfuerzos en el tiempo.
    - Frecuencia de cooperación vs. free-riding.
    - Comparación del "equilibrio emergente" con el óptimo social (si es calculable).
6. (Opcional) **Tener una fase evolutiva**:
    - En cada ronda de "torneo", las estrategias más exitosas se "replican" o se seleccionan para la siguiente fase.

- Observar si alguna estrategia dominante surge.

---

4. Cómo planear la estrategia de un agente

Cuando un estudiante diseña su **propio agente**, podría seguir estos pasos:

1. **Identificar objetivos**:
   - ¿Te interesa maximizar la nota a toda costa?
   - ¿Te interesa equilibrar calificación y horas de ocio?
   - ¿Estás dispuesto a castigar comportamientos egoístas?
2. **Definir tu modelo de "percepción"**:
   - ¿Qué información tendrás de las rondas anteriores? (¿Ves exactamente cuántas horas puso cada otro agente, o solo la calificación resultante?)
   - ¿Cómo procesas esa información? (¿Actualizas creencias sobre si es "confiable" cada uno?)
3. **Diseñar la regla de decisión** (estrategia):
   - Puede ser un **if-then** sencillo:
     "Si la calificación anterior fue baja, aumento mi esfuerzo para intentar subirla, de lo contrario me relajo".
   - O algo más sofisticado:
     "Utilizo un algoritmo de aprendizaje que, en función de (estado_actual), me sugiere cuántas horas poner".
4. **Probarla en escenarios simples**:
   - Empieza con un caso de 2 jugadores, T pequeño, y un f lineal.
   - Observa si tu agente obtiene más utilidad que un agente "constante".
   - Ajusta parámetros si ves comportamientos no deseados.
5. **Escalar y robustecer**:
   - Añade más jugadores, un T más grande, heterogeneidad en ω_e y α_e.
   - Observa cómo evoluciona la estrategia y si sigue siendo efectiva.

PROF

---

5. Visualización y análisis de resultados

Para entender el comportamiento de tu agente y el de los demás, puedes:

1. **Graficar la evolución de esfuerzos** a lo largo de las rondas:
   - Ver si los esfuerzos convergen a un nivel estable o fluctúan.
   - Identificar si alguien "colapsa" al no cooperar.
2. **Graficar la utilidad individual** por ronda o acumulada:
   - Distinguir quién obtiene mayor beneficio y en qué momento.

3. **Mapa de calor de decisiones**:
    - Si usas aprendizaje por refuerzo, puedes visualizar la **función de valor (Q-value)** que asigna a cada estado-acción, para ver cómo tu agente "piensa" internamente.
4. **Comparar agentes** en un ranking final:
    - ¿Cuál obtuvo la mayor utilidad promedio?
    - ¿Cuál alcanzó los mejores resultados sociales (suma total de utilidades)?
5. **Realizar un análisis de sensibilidad**:
    - Variar parámetros: α_e, ω_e, escala de f, etc.
    - Evaluar estabilidad de la estrategia ante estos cambios.

---

6. Conclusión: hacia un "torneo" colaborativo

- La propuesta final es que cada estudiante **programe** o **describa** su agente (con una regla de decisión clara y bien documentada).
- A continuación, se corren simulaciones donde se forman grupos aleatorios de agentes. Se repite para distintos escenarios y, al final, se obtiene un **"podio"** de estrategias más exitosas en cada contexto.
- Esta actividad sirve para entender en la práctica fenómenos de **cooperación**, **free-riding**, **castigo** y **colaboración** en **juegos repetidos** que reflejan la realidad de trabajar en equipo bajo presión, con distintas motivaciones y habilidades.

En resumen, **la clave** es diseñar y comparar **diferentes reglas de decisión**, observar sus resultados en múltiples condiciones y **analizar** cómo la dinámica de cooperación o explotación emerge de las interacciones repetidas. Cada estudiante puede **experimentar** con ideas sencillas, algoritmos de aprendizaje o sistemas de reputación para encontrar la estrategia que mejor se adapte a las variaciones del entorno. Con este enfoque, se lograría una **visión práctica** y experimental de la teoría subyacente al problema del free-rider y la cooperación en equipos.