1. Introducción

En un **juego repetido**, cada jugador elige acciones (puras o mixtas) en múltiples etapas (o rondas). Para encontrar los **equilibrios** (en el sentido de Equilibrio de Nash o, más específicamente, de Equilibrio Perfecto en Subjuegos), es útil distinguir dos grandes casos:

1. **Horizonte finito**: Se repite el juego base durante T etapas, con T conocido.
2. **Horizonte infinito** (o indefinido): Se repite el juego base de forma ilimitada, o con una probabilidad de continuar en cada etapa. Típicamente se introduce un factor de descuento δ.

A continuación, se presenta un **algoritmo general** (o procedimiento sistemático) para cada caso.

---

2. Algoritmo para juegos repetidos con horizonte finito

2.1 Especificación del juego base

1. **Identificar**:

   - Conjunto de jugadores N.
   - Conjunto de acciones A_i para cada jugador $i\in N$.
   - Funciones de utilidad $u_i: A_1 \times \cdots \times A_n \to \mathbb{R}$.

2. **Definir** la repetición T: número de etapas (rondas).

3. **(Opcional)** Incluir factor de descuento δ, si la recompensa futura se valora menos.

2.2 Procedimiento de **inducción hacia atrás** (Backward Induction)

El método estándar para juegos repetidos finitos es la **inducción hacia atrás**, que garantiza (bajo ciertas condiciones) un **Equilibrio Perfecto en Subjuegos**. El algoritmo paso a paso:

1. **Etapa final $\mathbf{T}$**:

   1. Considera únicamente el juego base en la última ronda T.
   2. **Resuelve** ese juego de una sola etapa (puede ser en estrategias puras o mixtas).
      - Formalmente, encuentra todos los Equilibrios de Nash del stage game en la ronda T.
      - Denota el (los) equilibrio(s) resultante(s) como $(\sigma_1^T,\dots,\sigma_n^T)$.
   3. Si hay múltiples equilibrios en la ronda T, se analizarán los que sean relevantes para la retroinducción.

2. **Etapa $\mathbf{T-1}$**:

   1. Toma en cuenta que el resultado de la ronda T (y sus estrategias) ya está caracterizado.
   2. **Define** el subjuego en la etapa $T-1$ considerando las utilidades de la ronda $T-1$ **más** la utilidad futura (ronda T) según la estrategia anticipada.

    3. **Resuelve** este subjuego (puro o mixto). Esto da la(s) estrategia(s) de equilibrio en la ronda $T-1$, teniendo en cuenta que todos los jugadores saben lo que sucederá en T.

3. **Continuar** el proceso recursivo:

    - Para $t = T-2, T-3,\dots,1$, en cada subjuego se asume que las estrategias (y payoffs) de las etapas posteriores ($t+1, t+2, \dots, T$) ya se han determinado.
    - **Resuelve** cada subjuego localmente para obtener las estrategias de cada jugador en la etapa t.

4. **Resultado**:

    - Al llegar a $t=1$, obtienes un **perfil de estrategias** $\sigma^* = (\sigma_1^,\dots,\sigma_n^)$ que describe la acción (pura o mezcla de acciones) de cada jugador en **todas** las etapas, condicionado a cada posible historial (si se requiere el refinamiento de subgame perfection).
    - Este σ^* constituye (en general) un **Equilibrio Perfecto en Subjuegos** y, por ende, también un Equilibrio de Nash del juego repetido.

Observación sobre mixtas vs. puras

- Cuando en la **última etapa** (juego base) se resuelve un Equilibrio de Nash mixto, se lleva esa solución a la penúltima etapa.
- El proceso es análogo si en alguna etapa se diera un equilibrio en estrategias mixtas. Se reemplazan los pagos deterministas por pagos **esperados** de la mezcla.

---

3. Algoritmo para juegos repetidos con horizonte infinito (o indefinido)

Para el caso **infinito**, no existe un "último paso" para hacer inducción hacia atrás. En su lugar, se usa un **procedimiento de "proponer y verificar"** (o *guess and check*), que también se puede interpretar dentro del marco de Equilibrio Perfecto en Subjuegos. A grandes rasgos:

3.1 Formalizar el juego repetido infinito

1. **Identificar** el stage game $(N, {A_i}, {u_i})$.
2. **Introducir** el factor de descuento $\delta \in (0,1)$.
    - El pago total de jugador i para un perfil de estrategias σ es:
    $$
    U_i(\sigma) ;=; \sum_{t=1}^{\infty} \delta^{,t-1},\mathbb{E}\bigl[u_i(a_i^t,a_{-i}^t)\bigr].
    $$

3.2 Proponer un **perfil de estrategias** candidato

Dado que las estrategias deben especificar **qué acción (o mezcla de acciones) toma cada jugador en cada historia** (potencialmente infinito), se suelen usar **familias de estrategias "condicionales"** (por ejemplo, disparo/grim trigger, tit-for-tat, castigo finito, etc.).

- **Ejemplo**: $\sigma = (\sigma_1,\sigma_2,\dots,\sigma_n)$ donde cada σ_i establece:

  > *"Coopera mientras todos cooperen; si alguien desvía, castígalo defectando por 5 rondas"*
  > o alguna otra regla de contingencia.

3.3 **Verificar incentivos** (condición de no desviación)

1. **Calcular** la utilidad esperada $U_i(\sigma)$ de cada jugador i **siguiendo** la estrategia σ^*.
2. Para cada jugador i, y para cada posible **desviación** σ_i' (pura o mixta) en cada historial, **comparar**:
   $
   U_i(\sigma_i^, *\sigma_{-i}^*) ;\ge; U_i(\sigma_i', \sigma_{-i}^*).
   $
3. Si **ningún** jugador puede aumentar su utilidad total (descontada) desviándose unilateralmente en ningún punto de la historia, entonces σ^* es un **Equilibrio de Nash** del juego repetido.
4. Si además se exige **perfección en subjuegos**, hay que verificar esta condición para **todos** los subjuegos iniciados tras cualquier historial.

En la práctica, se obtienen condiciones tipo:
$
\text{(beneficio de cooperar a largo plazo)} ;\ge; \text{(beneficio inmediato de desviarse)}.
$

3.3.1 Cálculo matemático de la utilidad

Para calcular la utilidad de un jugador σ_i dada la estrategia de los demás σ_{-i}:

- Se considera la **secuencia aleatoria** ${a_i^t, a_{-i}^t}*{t=1}^\infty$ inducida por $(\sigma_i, \sigma*{-i})$.
- Se acumula $\sum_{t=1}^\infty \delta^{t-1} u_i(a_i^t,a_{-i}^t)$.
- La comparación con otra estrategia σ_i' implica recomputar la secuencia resultante y su suma descontada.

PROF

3.4 (Opcional) Caracterizar el **conjunto de equilibria**

- En juegos repetidos infinitos, frecuentemente hay **múltiples** equilibrios (Folk Theorem).
- El "algoritmo" genérico en realidad se convierte en "probar" diferentes perfiles de estrategias y verificar si satisfacen la condición de no desviación.

---

4. Comentarios sobre estrategias mixtas

Tanto en el **horizonte finito** como en el **infinito**, la lógica de los algoritmos no cambia sustancialmente si permitimos que las estrategias sean **mixtas** en cada etapa. Lo que cambia es que:

1. En cada subjuego (o ronda), el **Equilibrio de Nash** del stage game puede consistir en mezclas de acciones.

2. El **perfil de estrategia** de un jugador en el juego repetido asigna una **distribución de probabilidad** sobre A_i, *dependiendo de la historia*.

El procedimiento sigue siendo el mismo, solo que en lugar de acciones puras, se resuelven los **best responses** mediante la **maximización de la utilidad esperada** dado el perfil de mezcla de los otros. Matemáticamente:

$
\max_{\sigma_i'} ; U_i(\sigma_i', \sigma_{-i})
\quad \text{donde } \sigma_i' : H \to \Delta(A_i).
$

---

5. Resumen formal del "Algoritmo" en pseudo-pasos

A modo de **síntesis**:

5.1 **Algoritmo para horizonte finito** $(T < \infty)$

1. **Entrada**:
   - $(N, {A_i}, {u_i})$ = juego base.
   - T = número de etapas.
   - (Opcional) δ = factor de descuento.
2. **Para** $t = T$ hasta 1 (en orden decreciente):
   1. Considerar el "subjuego" que inicia en la etapa t.
   2. Si $t = T$, resolver el stage game normal:
      - Hallar el (los) equilibrio(s) de Nash (puro/mixto) en una sola jugada.
   3. Si $t < T$, incorporar en los pagos la utilidad de la etapa t + la utilidad (descontada) que proviene de las estrategias en las etapas $(t+1,\dots,T)$ ya determinadas en pasos anteriores.
   4. Encontrar el (los) mejor(es) respuesta(s) de cada jugador y, de esa forma, el perfil de estrategias de equilibrio.
3. **Salida**:
   - Perfil de estrategias $\sigma^*(1), \sigma^*(2), \dots, \sigma^*(T)$, que describe qué hace cada jugador en cada etapa, (posiblemente condicionado a la historia) y satisface la condición de no desviación.

PROF

5.2 **Algoritmo para horizonte infinito** $(T = \infty)$

1. **Entrada**:
   - $(N, {A_i}, {u_i})$ = juego base.
   - δ = factor de descuento $\in (0,1)$.
2. **Proponer** una familia de estrategias $\Sigma^* = (\sigma_1^*,\dots,\sigma_n^*)$ con la lógica de castigos/recompensas (o alguna otra estructura).
3. **Para cada jugador** $i\in N$:
   - Calcular $U_i(\sigma_i^*, \sigma_{-i}^*)$.
   - Para toda desviación factible σ_i', calcular $U_i(\sigma_i', \sigma_{-i}^*)$.
   - Verificar la condición $;U_i(\sigma_i^*, \sigma_{-i}^*) \ge U_i(\sigma_i', \sigma_{-i}^*)$.

    - **Si** existe una desviación σ_i' que mejora la utilidad de i, Σ^* **no** es equilibrio. Ajustar o probar otra estrategia.
4. **Salida**:
    - Toda Σ^* que **no** permita mejoras unilaterales en ningún subjuego (historia posible) es un **Equilibrio Perfecto en Subjuegos**.

(En la práctica, se estudian "estrategias gatillo" o "tit-for-tat", etc., y se derivan condiciones sobre δ para que sean sostenibles.)

---

6. Conclusión

- El **algoritmo** de **inducción hacia atrás** resuelve de manera sistemática cualquier **juego repetido finito**.
- En **horizonte infinito**, el procedimiento habitual es de "**probar y verificar**" (check de incentivos), ya que no existe un paso final para la recursión.
- Las **estrategias mixtas** se incorporan simplemente cambiando la resolución de la etapa (o subjuego) a una búsqueda de Equilibrio de Nash en mezclas, pero el esqueleto del algoritmo permanece igual.
- Esta aproximación general funciona para definir y caracterizar tanto **Equilibrios de Nash** como **Equilibrios Perfectos en Subjuegos** en juegos repetidos.

De este modo, uno puede, de manera formal, **encontrar** (o al menos **verificar**) los equilibrios de un juego repetido, ya sea con horizonte finito o infinito, en estrategias puras o mixtas.