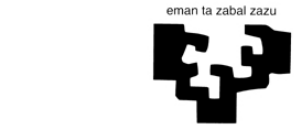


ZTF-FCT
Zientzia eta Teknologia Fakultatea
Facultad de Ciencia y Tecnología

# Mecanismo *Quid Pro Quo* a uno y dos niveles con aplicación a sistemas blockchain

Trabajo Fin de Grado
Grado en Matemáticas

Luis de la Pisa Arribas

Trabajo dirigido por
Josu Doncel Vicente

Leioa, 20 de junio de 2019

# Índice general

# Introducción.

El presente trabajo es fruto de un proyecto de investigación realizado gracias a la beca Ikasiker, ofrecida por el Gobierno Vasco al alumnado universitario que desea iniciarse en tareas de investigación. En mi caso, además del directordel trabajo, he colaborado con Agustín Santos y Antonio Fernández, investigadores del instituto IMDEA Networks en Madrid, para desarrollar un modelo matemático aplicable al registro de bloques en sistemas blockchain.

Los dos objetivos principales del proyecto han sido:

(i) Adaptar el mecanismo *Quid Pro Quo* ($QPQ$) que Santos y Fernández diseñaron originalmente para resolver el problema de asignación de tareas de manera óptima [1] al registro de bloques en una blockchain por orden preferente y demostrar que los resultados analíticos del mecanismo se mantienen. A lo largo del documento, llamamos a esta adaptación mecanismo $QPQ$ a un nivel.

(ii) Diseñar este último mecanismo con una nueva arquitectura que permite ejecutar el mecanismo con un menor intercambio de información. En este caso, denominamos al mecanismo $QPQ$ a dos niveles. Basándonos en la misma idea, proponemos como trabajo futuro la extensión del mecanismo a $m$ niveles, $m > 2$.

La motivación de adaptar el mecanismo *Quid Pro Quo* al registro de bloques en blockchain por orden preferente es doble: por un lado, mencionamos que la blockchain es una tecnología innovadora de registro de datos cuyas aplicaciones están creciendo constantemente y, por otro lado, que los requisitos que originalmente satisface el mecanismo *Quid Pro Quo* son también respetados en esta nueva contextualización, tal y como explicamos en la memoria.

En cuanto a la estructura del trabajo, presentamos en el primer capítulo una breve introducción a los sistemas blockchain y al problema de asignación de recursos. El registro de bloques en blockchain por orden preferente es un caso particular de éste último problema.

En el segundo y tercer capítulo presentamos y analizamos, respectivamente, el mecanismo $QPQ$ a un nivel. El mismo esquema es respetado para el mecanismo $QPQ$ a dos niveles en los dos siguientes capítulos. Finalmente, terminamos el trabajo presentando las conclusiones del proyecto y proponiendo al lector cuestiones abiertas relacionados con el mecanismo.

# Capítulo 1

# Asignación de recursos y blockchain

El objetivo de este primer capítulo es introducir brevemente el concepto de blockchain y presentar el problema de registro de bloques por orden de preferencia, el cual es un caso particular de la asignación de recursos. Por último, mencionamos las hipótesis generales en las que se basa el problema considerado.

## 1.1. Introducción a blockchain

La Tecnología de Registros Distribuidos (*DLT*, por sus siglas en inglés) hace referencia a un enfoque innovador y en creciente evolución de registro y compartimento de datos a través de múltiples *ledgers* (libros de contabilidad), los cuales tienen el mismo registro de datos y están controlados de forma colectiva por una red distribuida P2P [1] de computadores, los cuales se llaman nodos o jugadores [2].

Esta tecnología permite a sus miembros registrar, compartir y validar simultáneamente un *ledger* común en el que cada transacción de datos está registrada y no puede ser posteriormente modificada. Además, cabe destacar que los datos son administrados de forma descentralizada: son los propios nodos quienes tienen el control.

A día de hoy, el tipo de *DLT* más conocido es blockchain. Esta funciona usando criptografía y métodos algorítmicos para crear y legitimar la nueva información añadida al ledger, pero a diferencia de otros tipos de DLT, los datos se organizan en bloques que se unen formando una cadena digital.

[1] Una red P2P es una red de ordenadores que funcionan sin clientes ni servidores fijos; actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red.

Gran parte del éxito de blockchain es debido a que es la tecnología subyacente del Bitcoin, la moneda digital más usada en todo el mundo. Por ello, es frecuente asociar este tipo de DLT con las criptomonedas, a pesar de ser muchas las aplicaciones de esta tecnología, que asegura una manera más segura y rentable de crear y administrar bases de datos. Citamos, entre otros, los siguientes ámbitos donde se aplica blockchain [3] : almacenamiento en la nube distribuido, gestión de identidades, registro y verificación de datos, voto a través de Internet y servicios de notaría.

## 1.2. Asignación de recursos

Uno de los problemas de blockchain es decidir qué bloque añadimos a la cadena cuando los jugadores que la comparten realizan propuestas diferentes. Este problema es un caso particular de la asignación de recursos, que definimos formalmente a continuación.

**Definición 1.** El problema de asignación de recursos es una tupla $< \mathcal{R}, \mathcal{N}, \Theta >$ donde:

(i) $\mathcal{R} = \{r_1, r_2, ...\}$ es un conjunto ordenado de recursos.

(ii) $\mathcal{N} = \{n_1, ..., n_k\}$ es un conjunto de nodos o jugadores, de tamaño finito $k \in \mathbb{N}$.

(iii) $\Theta = \{\theta_j\}_{j=1}^{k}$ es un conjunto de variables aleatorias donde $\theta_j$ representa las preferencias del jugador $n_j$. Esta información es privada (solo es conocida por el jugador $n_j$). La preferencia de $n_j$ por un recurso $r$ se denota por $\theta_j(r)$.

En nuestro caso, los jugadores son los miembros que comparten la blockchain. El conjunto de recursos son las posiciones de la cadena en la que los jugadores añaden los bloques, las cuales se asignan de manera ordenada, es decir, si la última posición registrada en la cadena es $r_k$, la siguiente posición subastada para el registro del siguiente bloque será $r_{k+1}$.

Las preferencias de los jugadores por registrar su bloque en cada posición $r_k$ sirven parar realizar el registro por orden preferente, es decir, en cada $r_k$ se registrará el bloque propuesto por el jugador cuya preferencia por dicha posición es más alta. La preferencia (o urgencia) por registrar un bloque en la cadena puede variar según el contexto en el que utilicemos blockchain, tal y como justificamos en los siguientes ejemplos:

- Si la blockchain implementa una criptomoneda, como por ejemplo *bitcoin*, los jugadores son los usuarios de dicha moneda, cuyas preferencias por registrar sus bloques en la cadena dependen de las comisiones que reciban por tal registro.
- Consideremos una blockchain que ayuda a los bancos a reconciliar cuentas. En este caso, los bancos (jugadores) pueden tener mayor urgencia (preferencia) en registrar sus transacciones porque, hasta que ello no ocurra, puede haber otros procesos esperando.
- En el caso de implementar blockchain en micropagos (por ejemplo, la compra de un café), los jugadores serían los individuos que realizan el pago, y sus preferencias en registrar su bloque puede ser mayores si hasta que no quede registrado el pago no obtienen el producto por el que pagan (en este caso, el café).

## 1.3. Requisitos del modelo

El problema de registro de bloques a una blockchain cumple los siguientes requisitos.

- **Medida de utilidad abstracta.** La preferencia de un jugador por registrar su bloque en la cadena puede depender de diversos factores. Por ejemplo, en caso de que la blockchain implemente una criptomoneda, ante un mismo bloque un jugador puede primar el beneficio económico que le genera el hecho de registrarlo, mientras que otro puede dar más importancia a que queden registradas las transacciones implícitas en el bloque. Por ello, suponemos que cada jugador utiliza su propia métrica y unidades para medir su preferencia por los recursos.
- **No hay sistema de pago.** Asumimos que no es posible el pago entre los jugadores[2]. Para que lo fuera, sería necesario que los jugadores tuvieran la misma referencia monetaria (euro, dolar, ...). Sin embargo, dado que cada jugador mide su coste en sus propias unidades, es difícil encontrar dicha referencia.
- **Racionalidad de los jugadores.** El hecho de que los jugadores sean perfectamente racionales significa que estos son capaces de calcular siempre la mejor estrategia, lo cual conlleva a veces a resolver problemas muy complejos. Sin duda alguna, esto no es siempre posible y por ello suponemos que los jugadores están racionalmente limitados.

[2]Por ejemplo, un jugador podría pagar a otros para que declaren una preferencia menor por añadir su bloque, y así tener menos competencia.

- **No hay entidad central.** Tal y como hemos explicado anteriormente, la blockchain es administrada de forma descentralizada, es decir, funciona sin la presencia de una entidad central que indica a quien se le asigna cada recurso.

# Capítulo 2

# Introducción al mecanismo *QPQ* a un nivel

El mecanismo *Quid Pro Quo* ($QPQ$) presentado en [1] se diseña originalmente para repartir un conjunto de tareas entre un grupo de jugadores de forma que el trabajo total realizado por el grupo sea mínimo. Para cada tarea, los jugadores declaran los costes que les supone realizarla y finalmente esta es asignada al jugador cuyo coste es menor.

Aquí, en cambio, adaptamos el mecanismo para registrar bloques en una blockchain por orden de preferencia, con el objetivo de maximizar el beneficio total conseguido por el grupo de jugadores. De aquí en adelante, nos referimos a él como mecanismo $QPQ$ a un nivel.

A continuación, presentamos las restricciones que imponemos al modelo para poder aplicar el mecanismo y después nos centramos en explicar detalladamente el algoritmo.

## 2.1. Restricciones del mecanismo

En el capítulo anterior mencionábamos los requisitos generales del problema considerado. En adición a ellos, presentamos ahora unas condiciones suplementarias que garantizan un correcto funcionamiento del mecanismo $QPQ$ a un nivel en el contexto considerado.

Por un lado, suponemos que los jugadores están bien identificados y que el conjunto de todos ellos es invariante: no hay jugadores que dejen de participar en la blockchain, ni nuevos nodos que se unan. Asumimos también que cada vez que se vaya a registrar un bloque en la cadena, todos participan (es decir, todos realizan una propuesta) y que el jugador elegido por el

mecanismo es efectivamente quien registra el bloque en la cadena. Además, la comunicación entre ellos será

- **fiable.** En particular, en el algoritmo que describimos los jugadores intercambian sus preferencias por añadir su bloque a la cadena, y suponemos que estos valores son recibidos correctamente.
- **concurrente.** Suponemos que cada jugador envía su preferencia por añadir su bloque a la cadena antes de recibir los valores del resto de jugadores.

Por ello, suponemos que las preferencias de los jugadores no están correladas, es decir, que las variables aleatorias $\{\theta_j\}_{j=1}^{k}$ son independientes. También suponemos que estas son continuas.

## 2.2. Preferencias normalizadas

En un principio no podemos comparar las preferencias de los jugadores de forma numérica, ya que cada uno de ellos utiliza su propia métrica para medirlas. Para hacer posible la comparación, aplicamos la *Transformada Integral de Probabilidad (PIT)*[1] a las variables aleatorias que representan las preferencias de los jugadores.

**Definición 2.** *(Transformada Integral de Probabilidad, PIT).* Sea $\mathcal{X}$ una variable aleatoria continua con una función de distribución $F_{\mathcal{X}}$. Entonces, la $PIT$ define una nueva variable aleatoria $Y = F_{\mathcal{X}} \circ \mathcal{X}$.

Nuestro interés en la $PIT$ es que sigue una distribución uniforme en [0,1].

**Proposición 1.** Sea $\mathcal{X}$ una variable aleatoria con una función de distribución continua $F_{\mathcal{X}}$. Entonces, la variable aleatoria $\mathcal{Y}$ definida por la PIT, $\mathcal{Y} = F_{\mathcal{X}} \circ \mathcal{X}$, sigue una distribución uniforme normalizada.

*Demostración.* Por ser $F_{\mathcal{X}}$ una función de distribución, la variable aleatoria $\mathcal{Y}$ toma valores en el intervalo $[0, 1]$:

$$\mathcal{Y}(y) = (F_{\mathcal{X}} \circ \mathcal{X})(y) = F_{\mathcal{X}}(\mathcal{X}(y)) \in [0, 1], \ \ \forall y \in \mathcal{Y}.$$

Consideramos ahora la función de distribución $F_{\mathcal{Y}}$ asociada a $\mathcal{Y}$. Entonces,

[1] En inglés, Probability Integral Transformation.

- Si $y \leqslant 0$, $\quad F_{\mathcal{Y}}(y) = P[\mathcal{Y} \leqslant y] = 0.$

- Si $y \geqslant 1$, $\quad F_{\mathcal{Y}}(y) = P[\mathcal{Y} \leqslant y] = 1.$

- Si $y \in (0,1)$, $\quad F_{\mathcal{Y}}(y) = P[\mathcal{Y} \leqslant y] = P[F_{\mathcal{X}} \circ \mathcal{X} \leqslant y].$

  $F_{\mathcal{X}}$ es monótona creciente por ser función de distribución, luego podemos escribir

$$F_{\mathcal{Y}}(y) = P[\mathcal{X} \leqslant max\{F_{\mathcal{X}}^{-1}(y)\}] = F_{\mathcal{X}}[max\{x \mid F_{\mathcal{X}}(x) = y\}] = y.$$

Queda así demostrado que la función de distribución $F_{\mathcal{Y}}$ coincide con la uniforme normalizada y por lo tanto $\mathcal{Y}$ sigue dicha distribución. □

Motivados por este resultado, se define la preferencia normalizada del jugador $n_j \in \mathcal{N}$ como $\bar{\theta}_j := PIT(\theta_j)$. Por construcción, estas nuevas variables aleatorias siguen una distribución uniforme en (0,1) y en consecuencia es posible compararlas.

Además, como las preferencias de los jugadores son independientes, las normalizadas también lo son, por ser estas la composición de cada $\theta_j$ y su respectiva función de distribución que es continua, luego medible. Esto es un resultado inmediato que deducimos a partir del siguiente resultado relacionado con variables aleatorias (ver Teorema 2 de la Sección 4.2 en [4]).

**Proposición 2.** Sea $\{\mathcal{X}_{i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n_i\}$ un conjunto de variables aleatorias independientes sobre el espacio de probabilidad $(\Omega, \mathcal{F}, P)$. Entonces, las nuevas variables aleatorias

$$f_i(\mathcal{X}_{i,1}, \ldots, \mathcal{X}_{i,n_i}), \quad i = 1, ..., n,$$

donde $f_i : \mathbb{R} \to \mathbb{R}$ son funciones medibles, son independientes.

## 2.3. Mecanismo *Quid Pro Quo*

Por último en esta sección, presentamos en la Tabla 2.1 el algoritmo $QPQ$ a un nivel y a continuación explicamos el mecanismo en detalle.

Código para el jugador $n_j$ y la posición $r$ de la cadena.

1: Estimar el valor $\theta_j(r)$ (preferencia de $n_j$ por registrar su bloque en $r$)
2: Calcular el valor normalizado $\bar{\theta}_j(r) = (PIT(\theta_j))(r)$
3: Declarar $\dot{\theta}_j(r)$ como preferencia normalizada
4: Esperar a recibir los valores declarados $\dot{\theta}_i(r)$ del resto de jugadores.
5: **For all** $n_i \in \mathcal{N}$ **do**
6: **if not** $GoF\_\text{Test}(\dot{\theta}_i(r), Historic_i)$ **then**
7: $\ddot{\theta}_i(r) \leftarrow \text{Random}(\cup\{\dot{\theta}_x(r) \mid n_x \in \mathcal{N}, n_x \neq n_i\})$
8: **else**
9: $\ddot{\theta}_i(r) \leftarrow \dot{\theta}_i(r)$
10: **end if**
11: $Historic_i \leftarrow Historic_i \cup \{\ddot{\theta}_i(r)\}$
12: **end for**
13: Sea $d = argmax\{\ddot{\theta}_i(r) \mid n_i \in \mathcal{N}\}$
14: **if** $d = n_j$ **then**
15: El bloque de $n_j$ se registra en $r$
16: **else**
17: do nothing (El jugador $d$ es quien registra su bloque en $r$)
18: **end if**

Tabla 2.1: Algoritmo QPQ a 1 nivel

Observamos que cada vez que se subasta una posición $r$ de la cadena, los jugadores declaran un valor $\dot{\theta}_j(r)$ que representa su preferencia normalizada, pero la cual puede no coincidir con $\bar{\theta}_j(r)$, su verdadera preferencia normalizada por la $PIT$.

Como los jugadores envían los valores declarados antes de recibir las preferencias del resto (concurrencia) y estos son recibidos correctamente (fiabilidad), se mantiene la independencia de las preferencias declaradas.

### 2.3.1. Test de aceptación

Con el objetivo de incentivar a los jugadores a declarar sus verdaderas preferencias, implementamos en el mecanismo un test de aceptación que rechaza los valores declarados que no representen la verdadera preferencias normalizada. Este test es el mismo para todos los jugadores, de forma que ante un mismo valor declarado, todos lo aceptan o todos lo rechazan. También por esta razón, todos usan el mismo histórico de valores ($Historic_j$) para cada $n_j$: una lista que recoge las preferencias del jugador $n_j$ en las rondas anteriores y que es utilizada por el test de aceptación para comprobar si el

valor declarado representa o no la verdadera preferencia normalizada.

En cuanto al test utilizado en el mecanismo, este puede ser cualquier test de bondad de ajuste (test $GoF$, por sus siglas en inglés), como por ejemplo, el test $\chi^2$, o el de Kolmogorov-Smirnov. Para el estudio de los resultados analíticos, consideraremos que el test es perfecto, es decir, si un jugador declara su verdadera preferencia normalizada [2], entonces pasa el test; en caso contrario, el test detecta que el jugador es deshonesto.

En general, el test $GoF$ rechazará los valores declarados por un jugador $n_j$ que no siguen la distribución uniforme normalizada y/o que no son independientes respecto a los valores guardados en el histórico $Historic_j$.

### 2.3.2. Castigo

En principio, el valor declarado por $n_j$ es rechazado cuando no coincide con su preferencia normalizada, es decir, cuando el jugador decide ser deshonesto. Un posible castigo para los jugadores deshonestos podría ser, por ejemplo, la prohibición de poder registrar su bloque en esa ronda y tomar tan solo los valores declarados que han pasado el test de aceptación.

No obstante, cuando $n_j$ es deshonesto[3], el mecanismo genera para el jugador un nuevo valor que denotamos por $\hat{\theta}_j(r)$. Este se genera según la distribución uniforme normalizada y de forma pseudoaleatoria a partir de las preferencias declaradas en la ronda por el resto de jugadores ($\dot{\theta}_i(r)$, $i \neq j$). De este modo, todos regeneran el mismo nuevo valor para $n_j$, ya que si el valor fuera calculado por cada jugador de forma aleatoria, podría darse el caso de que, usando el mecanismo $QPQ$ a un nivel, los jugadores obtuvieran distintos ganadores para el registro del bloque en la cadena.

A pesar de que esta estrategia parece poco castigo para los jugadores deshonestos, evita que los jugadores honestos sean fuertemente castigados en caso de declarar su verdadera preferencia y que esta sea rechazada por el test $GoF$ [4]. Además, ello es un elemento crucial para asegurar que un jugador siempre obtendrá un mayor beneficio esperado declarando su verdadera preferencia normalizada.

[2] De ahora en adelante y cuando no haya lugar a confusión, nos referimos simplemente como preferencia normalizada.

[3] Decimos que un jugador es deshonesto cuando su valor declarado es rechazado por el test $GoF$. En caso contrario, decimos que es honesto.

[4] Aunque para el análisis del mecanismo consideremos el test $GoF$ perfecto, en la práctica el test puede fallar y considerar para un jugador honesto que su valor declarado no representa su preferencia normalizada.

Una vez que aplicamos a todos el test de aceptación, guardamos para cada jugador el valor $\ddot{\theta}_j(r)$, el cual coincidirá con el valor declarado cuando este pasa el test, y con el regenerado en la línea 7 del algoritmo (Tabla 2.1) en caso contrario. Son estos valores los que compara el mecanismo $QPQ$ a un nivel a la hora de determinar el jugador que registrará su bloque en la cadena, y por ello nos referimos a $\ddot{\theta}_j(r)$ como la preferencia de $n_j$ utilizada (o simplemente como la preferencia de $n_j$, cuando no hay lugar a confusión con $\theta_j(r)$).

Por último, el algoritmo determina el jugador cuya preferencia $\ddot{\theta}_j(r)$ es mayor, y cuyo bloque se registra en la blockchain en la posición considerada.

# Capítulo 3

# Análisis del mecanismo *QPQ* a un nivel

En este capítulo analizamos las propiedades matemáticas del mecanismo $QPQ$ a un nivel, teniendo en cuenta las suposiciones explicadas en la sección anterior. Recordamos que los resultados expuestos y sus correspondientes demostraciones son adaptaciones de los resultados propios del mecanismo explicado en [1], adaptados en este caso al contexto explicado: registro de bloques en una *blockchain* por orden preferente. Para ello, primero exponemos la notación empleada en este capítulo y en segundo lugar, presentamos los resultados analíticos obtenidos.

## 3.1. Notación

A lo largo del capítulo vamos a considerar el conjunto $\mathcal{N}$ de $k > 1$ jugadores. Las preferencias normalizadas (resp., declaradas) de cada jugador $n_j$ están representadas por las variables aleatorias $\bar{\theta}_j$ (resp., $\dot{\theta}_j$). Asimismo, las preferencias de $n_j$ utilizadas por el mecanismo para determinar quién efectuará el registro (línea 13 del algoritmo $QPQ$ a un nivel) se denotan por $\ddot{\theta}_j$. Recordamos que podemos referirnos a esta última variable simplemente como preferencia del jugador $n_j$.

En función de la verdadera preferencia normalizada, definiremos para cada jugador $n_j$ una variable aleatoria $\bar{\mathcal{P}}_j$ que representa su beneficio (normalizado)[1] por registrar su bloque en la cadena. Para cada posición $r \in \mathcal{R}$ tenemos

$$\bar{\mathcal{P}}_j(r) = \begin{cases} \bar{\theta}_j(r), & \text{si } n_j \text{ registra su bloque en } r. \\ 0, & \text{en caso contrario.} \end{cases}$$

[1]Nos referimos simplemente a esta variable como beneficio del jugador $n_j$.

Vemos que el beneficio de $n_j$ no depende del valor declarado $\dot{\theta}_j(r)$, sino de la preferencia normalizada $\bar{\theta}_j(r)$, la cual solo conoce el propio jugador. También podemos observar que cuanto mayor es $\bar{\theta}_j(r)$, mayor es el beneficio que supone a $n_j$ registrar su bloque en la posición $r$.

Comentamos que, en caso de que el valor declarado por $n_j$ para registrar su bloque en $r$ no pase el test de aceptación, denotamos el valor regenerado por el mecanismo por $\hat{\theta}_j(r)$. Análogamente, también en este caso denotamos el beneficio de $n_j$ asociado a $r$ por $\hat{P}_j(r)$.

De esta manera, diferenciamos entre el beneficio de $n_j$ cuando su preferencia declarada pasa el test de aceptación ($\bar{\mathcal{P}}_j$) y el correspondiente cuando la preferencia declarada es rechazada por el test ($\hat{\mathcal{P}}_j$). Como para el análisis del mecanismo consideramos el test perfecto, estas dos variables aleatorias representarán el beneficio de $n_j$ cuando es honesto y deshonesto, respectivamente.

## 3.2. Resultados principales

Esta sección está dividida en diferentes apartados. En el primero, demostramos que el mecanismo QPQ a un nivel es óptimo cuando todos los jugadores son honestos y en los siguientes, probamos resultados que nos acercan a demostrar que la mejor estrategia de un jugador para obtener el mayor beneficio posible es ser honesto. Es decir, queremos demostrar, $\forall j \in \mathcal{N}$, que

$$E[\bar{\mathcal{P}}_j] > E[\hat{\mathcal{P}}_j].$$

### 3.2.1. Mecanismo óptimo para jugadores honestos

En primer lugar, probamos que el algoritmo es óptimo en el sentido de que, si todos los jugadores son honestos, el beneficio total generado para todos ellos es maximizado.

**Proposición 3.** Si todos los jugadores son honestos, no existe ningún mecanismo $M$ distinto del mecanismo $QPQ$ a un nivel tal que, para un conjunto de posiciones $\mathcal{R}$ de la cadena, verifique

$$E[\sum_{j=1}^{n} \bar{\mathcal{P}}_j^M] > E[\sum_{j=1}^{n} \bar{\mathcal{P}}_j],$$

donde $\bar{\mathcal{P}}_j^M$ es la variable aleatoria que representa el beneficio (normalizado) del jugador $n_j$ asociado al mecanismo $M$.

*Demostración.* Supongamos que todos los jugadores son honestos y que existe un mecanismo M tal que

$$E[\sum_{j=1}^{n} \bar{\mathcal{P}}_j^M] > E[\sum_{j=1}^{n} \bar{\mathcal{P}}_j]. \tag{3.1}$$

En este caso, debe existir al menos una posición $r$ de la cadena tal que

$$\sum_{j=1}^{n} \bar{\mathcal{P}}_j^M(r) > \sum_{j=1}^{n} \bar{\mathcal{P}}_j(r).$$

Como en cada posición $r$ de la cadena solo puede ser registrado un bloque, todos los jugadores tienen un beneficio nulo esperado, excepto aquél que registra el suyo. Considerando que usando los mecanismos $M$ y $QPQ$ a un nivel los jugadores que registran su bloque en $r$ son, respectivamente, $n_j$ y $n_i$, podemos escribir

$$\bar{\mathcal{P}}_j^M(r) > \bar{\mathcal{P}}_i(r).$$

Como todos los jugadores son honestos, el jugador que registra su bloque en $r$ usando el mecanismo $QPQ$ a un nivel es aquel cuya preferencia normalizada es mayor. En consecuencia, no hay ningún jugador cuyo beneficio por registrar su bloque en $r$ pueda ser mayor y por ello no existe ningún mecanismo que cumpla (3.1). □

### 3.2.2. Distribución de las preferencias normalizadas

Un resultado básico que utilizamos a lo largo de esta sección es el que demostramos en este apartado. Argumentamos que las preferencias que utiliza el mecanismo $QPQ$ a un nivel para determinar el jugador cuyo bloque será registrado en la cadena (línea 13 del algoritmo $QPQ$ a un nivel) siguen una distribución uniforme en (0,1), y además son independientes. Para ello, diferenciamos dos casos:

- **Los jugadores son honestos.** En este caso, los valores declarados coinciden con las verdaderas preferencias normalizadas. Consecuentemente y por ser el test perfecto, tenemos que $\ddot{\theta}_j(r) = \bar{\theta}_j(r)$, $\forall n_j \in \mathcal{N}$. Por tanto, se mantiene la distribución uniforme normalizada y la independencia propia de las verdaderas preferencias normalizadas.

- **Los jugadores son deshonestos.** Cuando el valor declarado no coincide con la verdadera preferencia normalizada, el mecanismo genera para el jugador deshonesto un nuevo valor según la distribución uniforme normalizada. Además, este nuevo valor está generado de forma pseudoaleatoria por lo que se conserva la independencia respecto a las preferencias $\ddot{\theta}_j(r)$ del resto de jugadores, coincidan o no con su verdadera preferencia normalizada.

Comentamos que al considerar el test perfecto, no existirán jugadores deshonestos que pasen el test de aceptación. En consecuencia, obtenemos el siguiente resultado.

**Proposición 4.** Las preferencias[2] de los jugadores, $\ddot{\theta}_j$, $j = 1, ..., k$, son independientes y uniformemente distribuidas en (0,1).

### 3.2.3. Jugador agregado

Un aspecto que simplifica el análisis y que además es compatible con el modelo, es la idea de jugador agregado (*aggregated player*). Consideramos que cada jugador $n_j$ compite contra otro ficticio que agrupa las preferencias del resto. Llamamos a este jugador ficticio jugador agregado y, siguiendo la notación de teoría de juegos [5], lo denotamos por $n_{-j}$.

Definimos también la preferencia utilizada de $n_j$ por agregar su bloque en una posición $r$ de la cadena como la preferencia utilizada máxima de los jugadores en la agregación. Denotamos este valor por $\ddot{\theta}_{-j}(r)$, el cual puede expresarse como sigue:

$$\ddot{\theta}_{-j}(r) := max\{\bar{\theta}_i(r) \mid 1 \leqslant i \leqslant k,\ i \neq j\}.$$

El objetivo de este apartado es calcular la distribución que siguen las preferencias (utilizadas) del jugador agregado. Para ello, presentamos a continuación el siguiente resultado.

**Proposición 5.** Sean $\mathcal{X}_1, ..., \mathcal{X}_k$, $k > 1$, un conjunto de variables aleatorias independientes que siguen, respectivamente, una distribución $Beta(p_j, 1)$, $p_j \in \mathbb{N}$, $\forall j \in \{1, ..., k\}$. Entonces, la variable aleatoria

[2]Las preferencias utilizadas por el mecanismo $QPQ$ a un nivel para determinar el jugador cuyo bloque se registrará en la cadena.

$$\mathcal{X} := \text{máx}\{\mathcal{X}_j \mid 1 \leq j \leq k\}$$

sigue una distribución $Beta(\sum_{j=1}^{k} p_j, 1)$.

*Demostración.* Primero, mencionamos que $\mathcal{X}$ es una variable aleatoria continua, por ser el resultado de aplicar la función máximo (que es medible) a un conjunto de $k > 1$ variables aleatoria continuas $\{\mathcal{X}_1, ..., \mathcal{X}_k\}$.

Consideramos entonces $F_{\mathcal{X}}$, la función de distribución de $\mathcal{X}$, y calculamos su valor para cualquier $x \in \mathbb{R}$.

$$F_{\mathcal{X}}(x) = P[\mathcal{X} \leqslant x] =$$

$$= P[\text{máx}\{\mathcal{X}_j \mid 1 \leq j \leq k\} \leqslant x] =$$

$$= P[\mathcal{X}_1 \leqslant x,\ ...,\ \mathcal{X}_k \leqslant x].$$

Por hipótesis, estas variables son independientes, luego

$$F_{\mathcal{X}}(x) = \prod_{j=1}^{k} P[\mathcal{X}_j \leqslant x] = \prod_{j=1}^{k} \left( \int_{-\infty}^{x} f_j(y) dy \right),$$

donde $f_j(y)$ es la función de densidad asociada a $\mathcal{X}_j$, la cual sigue una distribución $Beta(p_j, 1)$, $\forall j \in \{1, ..., k\}$. Esto es,

$$f_j(y) = \begin{cases} p_j \cdot y^{p_j - 1}, & \text{si } y \in (0, 1). \\ \\ 0, & \text{en caso contrario.} \end{cases}$$

Dependiendo del valor de $x$, diferenciamos tres casos:

- $F_{\mathcal{X}}(x) = \prod_{j=1}^{k} (\int_{-\infty}^{x} 0 \, dy) = 0,$ si $x \leqslant 0$.
- $F_{\mathcal{X}}(x) = \prod_{j=1}^{k} (\int_{0}^{x} p_j \cdot y^{p_j - 1} \, dy) = x^{(\sum_{j=1}^{k} p_j)},$ si $x \in (0, 1)$.
- $F_{\mathcal{X}}(x) = \prod_{j=1}^{k} (\int_{(-\infty, 0) \cup (1, x)} 0 \, dx + \int_{0}^{1} p_j \cdot y^{p_j - 1} \, dy) = 1,$ si $x \geqslant 1$.

Derivando $F_{\mathcal{X}}$ respecto a $x$, obtenemos la función de densidad de $\mathcal{X}$, $f_{\mathcal{X}}$:

$$f_{\mathcal{X}}(x) = \begin{cases} \left(\sum_{j=1}^{k} p_j\right) \cdot x^{\left(\sum_{j=1}^{k} p_j\right)-1}, & \text{si } x \in (0,1). \\ \\ 0, & \text{en caso contrario.} \end{cases}$$

Por último, recordamos que la función de densidad $f_\beta$ asociada a una distribución $Beta(p,q)$ es:

$$f_\beta(x) = \begin{cases} \frac{1}{\beta(p,q)} \cdot x^{p-1} \cdot (1-x)^{q-1}, & \text{si } x \in (0,1). \\ \\ 0, & \text{en caso contrario.} \end{cases}$$

En concreto, cuando $p = \sum_{j=1}^{k} p_j$ y $q = 1$ tenemos:

(i) Si $x \in (0,1)$,

$$f_\beta(x) = \frac{1}{\beta\left(\sum_{j=1}^{k} p_j, 1\right)} x^{\left(\sum_{j=1}^{k} p_j\right)-1} =$$

$$= \frac{\Gamma(1+\sum_{j=1}^{k} p_j)}{\Gamma(1)\Gamma(\sum_{j=1}^{k} p_j)} x^{\left(\sum_{j=1}^{k} p_j\right)-1} =$$

$$= \left(\sum_{j=1}^{k} p_j\right) x^{\left(\sum_{j=1}^{k} p_j\right)-1}.$$

(ii) En caso contrario,

$$f_\beta(x) = 0.$$

Como $f_{\mathcal{X}}$ es igual a $f_\beta$ en este último caso, afirmamos finalmente que $\mathcal{X}$ sigue una distribución $Beta(\sum_{j=1}^{k} p_j, 1)$. □

La distribución que siguen las preferencias de los jugadores agregados se calculan fácilmente a partir de esta proposición.

**Corolario 1.** Las preferencias $\ddot{\theta}_{-j}$ del jugador agregado $n_{-j}$ siguen una distribución $Beta(k-1,1)$.

*Demostración.* Como la preferencia del jugador agregado es la mayor de las preferencias de los jugadores que forman la agregación, podemos expresar $\ddot{\theta}_{-j}$ como sigue:

$$\ddot{\theta}_{-j} := max\{\ddot{\theta}_i \mid 1 \leqslant i \leqslant k,\ i \neq j\}.$$

Por la Proposición 4, sabemos que las variables aleatorias $\ddot{\theta}_i$, $i \neq j$, son independientes y uniformes en $(0, 1)$. Además, esta distribución equivale con la $Beta(1, 1)$, pues su función de distribución es

$$f_\beta(y) = \begin{cases} \frac{1}{\beta(1,1)} \cdot y^0 \cdot (1-y)^0 = 1, & \text{si } y \in (0,1). \\ \\ 0, & \text{en caso contrario.} \end{cases}$$

Por construcción y aplicando ahora la Proposición 5, podemos afirmar que $\ddot{\theta}_{-j}$ sigue una distribución $Beta(k-1, 1)$. □

### 3.2.4. Estrategias de los jugadores

En esta sección calculamos el beneficio esperado por un jugador diferenciando el caso honesto (el jugador declara sus verdaderas preferencias normalizadas) y el deshonesto (el jugador declara cualquier otro tipo de valores). Una vez realizado los cálculos, comparamos el beneficio esperado según el caso para demostrar que estos siempre esperan un mayor beneficio cuando son honestos, independientemente de las estrategias[3] del resto.

Para ello, es importante tener en cuenta que cuando se *subasta* una posición $r$ de la cadena, un jugador $n_j$ obtiene un beneficio no nulo cuando es su propio bloque el que se registra. Esto ocurre cuando la preferencia utilizada del jugador agregado $n_{-j}$ es menor que la del propio $n_j$. Este hecho se expresa como sigue:

$$\ddot{\theta}_{-j} < \ddot{\theta}_j(r).$$

Calculamos a continuación el beneficio esperado por un jugador honesto.

**Proposición 6.** El beneficio esperado de $n_j$ cuando es honesto es

$$E[\bar{P}_j] = \frac{1}{k+1},$$

[3]Las estrategias de un jugador se reducen básicamente a declarar su verdadera preferencia normalizada (ser honesto) o cualquier otro valor (ser deshonesto).

para cualquier estrategia del resto de jugadores.

*Demostración.* Supongamos que $n_j$ es honesto. Entonces, para cualquier posición $r \in \mathcal{R}$, tenemos que la preferencia $\ddot{\theta}_j(r)$ coincide con la verdadera preferencia normalizada $\bar{\theta}_j(r)$. El beneficio por registrar su bloque en dicha posición es entonces

$$\bar{P}_j(r) = \begin{cases} \bar{\theta}_j(r), & \text{si } \ddot{\theta}_{-j} < \bar{\theta}_j(r). \\ \\ 0, & \text{en caso contrario.} \end{cases}$$

Asimismo, el beneficio esperado por $n_j$ es

$$E[\bar{P}_j] = \int_{\mathbb{R}} x \cdot \left( f(x) \cdot P[\ddot{\theta}_{-j} < x] + 0 \cdot P[\ddot{\theta}_{-j} \geqslant x] \right) \, dx,$$

donde $f(x)$ es la función de densidad asociada a $\bar{\theta}_j$, la cual sigue la distribución uniforme normalizada. Entonces,

$$E[\bar{P}_j] = \int_0^1 x \cdot P[\ddot{\theta}_{-j} < x] \, dx.$$

Por el Corolario 1, las preferencias del jugador agregado siguen una distribución $Beta(k-1, 1)$, independientemente de la honestidad de los jugadores que lo forman. Por consiguiente, el beneficio esperado por $n_j$ cuando es honesto es, para cualquier estrategia del resto de jugadores, el siguiente:

$$E[\bar{P}_j] = \int_0^1 x \cdot \left( \int_0^x (k-1) y^{k-2} \, dy \right) \, dx =$$

$$= \int_0^1 x \cdot \left( x^{k-1} \right) \, dx =$$

$$= \frac{1}{k+1}.$$

□

Sin embargo, el beneficio esperado por $n_j$ cambia cuando este decide ser deshonesto, tal y como demostramos en la siguiente proposición.

**Proposición 7.** El beneficio esperado de $n_j$ cuando es deshonesto es

$$E[\hat{P}_j] = \frac{1}{2k},$$

para cualquier estrategia del resto de jugadores.

*Demostración.* Supongamos que $n_j$ es deshonesto. Ahora, para cualquier posición $r \in \mathcal{R}$, su beneficio por registrar su bloque en dicha posición es

$$\hat{P}_j(r) = \begin{cases} \bar{\theta}_j(r), & \text{si } \ddot{\theta}_{-j} < \hat{\theta}_j(r), \\ \\ 0, & \text{en caso contrario,} \end{cases}$$

donde $\hat{\theta}_j(r)$ representa el valor regenerado (de acuerdo a la distribución uniforme normalizada) por el mecanismo $QPQ$ a un nivel.

En este caso, el beneficio esperado por $n_j$ es

$$E[\hat{\mathcal{P}}_j] = \int_{\mathbb{R}} x \cdot \Big( f(x) \cdot P[\ddot{\theta}_{-j} < z] \Big)\ dx,$$

donde $f(x)$ es la función de densidad asociada a $\bar{\theta}_j$, la cual sigue la distribución uniforme normalizada, y $z$ representa el valor regenerado del jugador, que sigue la misma distribución. Se sigue que

$$E[\hat{\mathcal{P}}_j] = \int_0^1 x \cdot P[\ddot{\theta}_{-j} \leqslant z]\ dx.$$

De nuevo, el Corolario 1 nos garantiza que las preferencias de $n_{-j}$ siguen una distribución $Beta(k-1, 1)$. Por ello, podemos seguir que el beneficio esperado por $n_j$ cuando es deshonesto es, para cualquier estrategia del resto de jugadores,

$$E[\hat{\mathcal{P}}_j] = \int_0^1 x \cdot \left( \int_0^1 \left( \int_0^z (k-1) y^{k-2}\ dy \right) dz \right)\ dx =$$

$$= \int_0^1 x \cdot \left( \frac{1}{k} \right)\ dx = \frac{1}{2k}.$$

Y así queda probado que cuando el jugador $n_j$ es deshonesto, entonces

$$E[\hat{P}_j] = \frac{1}{2k}.$$

□

Comparando estas dos últimas proposiciones deducimos que usando el mecanismo QPQ a un nivel, los jugadores obtienen mayor beneficio cuando son honestos que cuando deciden mentir sobre sus preferencias.

**Corolario 2.** Para cualquier jugador $n_j \in \mathcal{N}$ se cumple la siguiente desigualdad:

$$E[\hat{P}_j] < E[\bar{P}_j].$$

*Demostración.* Supongamos por reducción al absurdo que $E[\hat{P}_j] \geqslant E[\bar{P}_j]$. Por las Proposiciones 6 y 7, escribimos

$$\frac{1}{2k} \geqslant \frac{1}{k+1} \iff k+1 \geqslant 2k \iff 1 \geqslant k.$$

Hemos caído en contradicción, ya que el número de jugadores $k$ es mayor que 1 por hipótesis. Por lo tanto, se tiene que

$$E[\hat{P}_j] < E[\bar{P}_j].$$

□

# Capítulo 4

# Introducción al mecanismo *QPQ* a dos niveles

El mayor problema del mecanismo QPQ a un nivel está relacionado con su implementabilidad en casos reales. En general, en los contextos donde se utiliza *blockchain* (ver Sección 1.2) el número de jugadores es muy grande y a veces es inviable intercambiar las preferencias declaradas entre todos los jugadores. Veamos un ejemplo.

**Ejemplo 1.** Suponemos una blockchain compartida por mil jugadores en la que se aplica el mecanismo $QPQ$ a un nivel para determinar qué bloque se añade a la cadena. Como cada jugador tiene que enviar al resto su preferencia declarada, el número total de mensajes intercambiados es en este caso

$$1000 \cdot 999 = 999\,000.$$

Es decir, en el tercer paso del algoritmo $QPQ$ a un nivel (ver Tabla 2.1) tenemos que esperar a recibir un total de 999 000 mensajes.

Diseñamos en este capítulo el mecanismo $QPQ$ con una una nueva arquitectura, el cual llamamos mecanismo QPQ a dos niveles. Este es una extensión natural del mecanismo QPQ a un nivel, y ambos se utilizan para el registro de bloques en una *blockchain* por orden preferente. Sin embargo, en comparación con el mecanismo $QPQ$ a un nivel, el nuevo diseño precisará de un menor intercambio de mensajes para que el mecanismo sea ejecutado. Ello nos permite aplicar el mecanismo $QPQ$ a dos niveles en situaciones donde, por ser el número de jugadores muy alto, el mecanismo QPQ a un nivel no puede.

## 4.1. Restricciones y fundamentos del mecanismo

Explicamos a continuación, junto a las restricciones consideradas para garantizar un funcionamiento correcto, la idea en la que se fundamenta el mecanismo QPQ a dos niveles.

En primer lugar, aplicamos las restricciones propias del mecanismo QPQ a un nivel explicadas en el Capítulo 2: los jugadores están bien identificados, la comunicación entre ellos es fiable y concurrente y en cada ronda todos publican sus preferencias, las cuales no están correladas entre sí.

Por otro lado, explicamos la idea en la que se basa este nuevo mecanismo:

Primero, se considera un conjunto de jugadores que comparten una *blockchain* y se divide en $n > 1$ subconjuntos o clusters,

$$\mathcal{N}_1, \ldots, \mathcal{N}_n,$$

de tamaño $k_1 > 1, \ldots, k_n > 1$, respectivamente. Consideramos que estos son fijos: cada jugador pertenece a un único mismo cluster en todas las rondas; además, su estructura y tamaños los suponemos conocidos por todos los jugadores.

Para determinar el ganador de cada cluster, es decir, el jugador del cluster cuya preferencia por registrar su bloque es mayor, aplicamos el mecanismo QPQ a un nivel (es decir, el mecanismo explicado en la sección anterior) a cada $\mathcal{N}_j$ y formamos un nuevo conjunto

$$\mathcal{N}^{(1)} = \{n_1, \ldots, n_n\}.$$

donde $n_1, ..., n_n$ representan los ganadores de los clusters $\mathcal{N}_1, ..., \mathcal{N}_n$, respectivamente.

Finalmente, el jugador cuyo bloque se registra en la cadena se determina aplicando el algoritmo QPQ a un nivel a los ganadores de los clusters, es decir, el conjunto $\mathcal{N}^{(1)}$.

## 4.2. Algoritmo QPQ a dos niveles

Para el análisis matemático, consideramos el conjunto de jugadores $\mathcal{N}$ dividido en $n > 1$ clusters $\mathcal{N}_1, ..., \mathcal{N}_n$ de tamaño $k_1 > 1, ..., k_n > 1$, respectivamente. Es decir,

$$\mathcal{N} = \cup_{j=1}^{k} \mathcal{N}_j.$$

donde $\mathcal{N}_j = \{n_{j,1}, ..., n_{j,k_j}\}$, $\forall j \in \{1, ..., n\}$.

Observamos que $\forall i \in \{1, ..., k_j\}$, el jugador $n_{j,i}$ pertenece a un único cluster $\mathcal{N}_j$, y que el número total de jugadores es $k = \sum_{j=1}^{n} k_j$.

A continuación, presentamos el algoritmo QPQ a dos niveles y seguidamente explicamos el mecanismo en detalle.

| Código para el jugador $n_{j,i}$ y la posición $r$ de la cadena. |
|---|
| 1: Estimar el valor $\theta_{j,i}(r)$ (preferencia de $n_{j,i}$ por registrar su bloque en $r$) |
| 2: Calcular el valor normalizado $\bar{\theta}_{j,i}(r) = (PIT(\theta_{j,i}))(r)$ |
| 3: Declarar $\dot{\theta}_{j,i}(r)$ como preferencia normalizada |
| 4: Esperar a recibir los valores declarados $\dot{\theta}_{j,v}(r)$ del resto de jugadores en $\mathcal{N}_j$ |
| 5: **For all** $n_{j,v} \in \mathcal{N}_j$ **do** |
| 6: **if not** $GoF\_\text{Test}(\dot{\theta}_{j,v}(r), Historic_{j,v})$ **then** |
| 7: $\ddot{\theta}_{j,v}(r) \leftarrow \text{Random}(\cup\{\dot{\theta}_{j,y}(r) \mid n_{j,y} \in \mathcal{N}, n_{j,y} \neq n_{j,v}\})$ |
| 8: **else** |
| 9: $\ddot{\theta}_{j,v}(r) \leftarrow \dot{\theta}_{j,v}(r)$ |
| 10: **end if** |
| 11: $Historic_{j,v} \leftarrow Historic_{j,v} \cup \{\ddot{\theta}_{j,v}(r)\}$ |
| 12: **end for** |
| 13: Sea $n_j = argmax\{\ddot{\theta}_{j,v}(r) \mid n_{j,v} \in \mathcal{N}_j\}$ |
| 14: **if** $n_j = n_{j,i}$ **then** |
| 15: Esperar a recibir las preferencias de $n_u = argmax\{\ddot{\theta}_{u,v}(r) \mid 1 \leqslant v \leqslant k_u\}$, $u \neq j$ |
| 16: Sea $\mathcal{N}^{(1)} = \cup_{u=1}^{n}\{n_u\}$ |
| 17: Sea $d = argmax\{\ddot{\theta}_u(r) \mid n_u \in \mathcal{N}^{(1)}\}$ |
| 18: **if** $d = n_{j,i}$ **then** |
| 19: El bloque de $n_{j,i}$ se registra en $r$ |
| 20: **else** |
| 21: do nothing (El jugador $d$ es quien registra su bloque en $r$) |
| 22: **end if** |
| 23: **else** |
| 24: do nothing (El jugador $n_j$ es el jugador de $\mathcal{N}_j$ con mayor preferencia) |
| 25: **end if** |

Tabla 4.1: Algoritmo QPQ a dos niveles

En primer lugar se aplica al cluster $\mathcal{N}_j$ al que pertenece $n_{j,i}$ los mismos pasos del algoritmo QPQ a un nivel: el jugador declara un valor $\dot{\theta}_{j,i}$ que en principio representa su verdadera preferencia normalizada ($\bar{\theta}_{j,i}$ ). Para com-

probarlo, aplicamos el test $GoF$ a las preferencias declaradas por todos los jugadores del cluster. Dicho test[1] se basa en los valores declarados en rondas anteriores de cada jugador $n_{j,i}$, respectivamente, los cuales están guardados en una lista $Historic_{j,i}$.

Consideramos aquí también que el test es perfecto: si $n_{j,i}$ declara su verdadera preferencia normalizada (es honesto), entonces pasa el test; en caso contrario, el test detecta que el jugador es deshonesto y regenera un nuevo valor $\hat{\theta}_{j,i}(r)$. Este es generado según la distribución uniforme normalizada y de forma pseudoaleatoria a partir de las preferencias declaradas en la ronda por el resto de jugadores ($\dot{\theta}_{j,v}(r)$, $v \neq i$).

En ambos casos, el valor de $n_{j,i}$ que ha pasado el test o ha sido regenerado, según el caso, se guarda como $\ddot{\theta}_{j,i}(r)$ en $Historic_{j,v}$.

En la línea 13, el algoritmo calcula el jugador en $\mathcal{N}_j$ cuya preferencia $\ddot{\theta}_{j,i}(r)$ por registrar su bloque en $r$ es mayor. Si no es $n_{j,i}$, su bloque no será registrado en $r$ y el algoritmo se termina para el jugador. En caso de serlo, el algoritmo continua considerando $\mathcal{N}^{(1)}$ (el conjunto que agrupa los ganadores de los $n$ clusters) y las preferencias utilizadas de los jugadores que forman dicho conjunto: $max\{\ddot{\theta}_{u,v}(r) \mid 1 \leqslant v \leqslant k_u\}$, $u = 1, ..., n$.

Por último, el algoritmo determina el jugador de $\mathcal{N}^{(1)}$ cuya preferencia por registrar su bloque en $r$ es mayor, y es dicho jugador quien finalmente registra su bloque en la cadena. Observamos que no hemos incluido a este segundo nivel un test de aceptación. La razón es que el test $GoF$ considerado en la línea 13 del algoritmo $QP$ a dos niveles es perfecto: este acepta los valores declarados por los jugadores honestos y rechaza los correspondientes a los deshonestos.

Vemos que, en comparación con el mecanismo QPQ a un nivel, este nuevo mecanismo precisa de menos mensajes para ser ejecutado.

**Ejemplo 2.** En una blockchain compartida por mil jugadores, cada vez que queramos decidir qué bloque añadir a la cadena utilizando el mecanismo $QPQ$ a un nivel, se precisa el intercambio de 999 000 mensajes, tal y como hemos calculado en el Ejemplo 1.

En cambio, si dividimos el conjunto de jugadores en cincuenta clusters de veinte jugadores cada uno y aplicamos el mecanismo $QPQ$ a dos niveles, el número total de mensajes intercambiados es entonces

[1] Al igual que para el mecanismo $QPQ$ a un nivel, podemos utilizar cualquier test de bondad de ajuste, como por ejemplo el test $\chi^2$ o el de Kolmogorov-Smirnov.

$$50 \cdot (20 \cdot 19) + 50 \cdot 49 = 21\,450.$$

Es decir, en cada cluster cada jugador comunica su preferencia declarada solo a los jugadores de su cluster. Después, los ganadores de los clusters reciben las preferencias utilizadas (coincidan con la verdadera preferencia normalizada o hayan sido regeneradas por no pasar el test de aceptación) de los ganadores de los clusters restantes.

Podemos calcular que utilizando el mecanismo $QPQ$ a dos niveles, el número de mensajes intercambiados para el registro de un bloque es, aproximadamente, cincuenta veces menor que usando el mecanismo a un nivel. Ello influye muy positivamente en el tiempo de ejecución del algoritmo a dos niveles.

# Capítulo 5

# Análisis del mecanismo *QPQ* a dos niveles

El objetivo principal de este capítulo es demostrar resultados análogos a los probados para el mecanismo $QPQ$ a un nivel en el Capítulo 3. Siguiendo el mismo esquema, exponemos primero la notación empleada a lo largo del capítulo y en segundo lugar, presentamos los resultados analíticos realizados sobre el mecanismo.

## 5.1. Notación

A lo largo del capítulo vamos a considerar el conjunto $\mathcal{N}$ de $k > 1$ jugadores tal y como explicamos en la Sección 4.2. Las preferencias normalizadas (resp., declaradas) de cada jugador $n_{j,i}$ están representadas por las variables aleatorias $\bar{\theta}_{j,i}$ (resp., $\dot{\theta}_{j,i}$).

Las preferencias de $n_j$ utilizadas por el mecanismo $QPQ$ a dos niveles para determinar el ganador del cluster[1] (línea 13 del algoritmo $QPQ$ a un nivel) se denotan en cambio por $\ddot{\theta}_{j,i}$.

Al igual que para el mecanismo $QPQ$ a un nivel, definiremos para cada jugador $n_{j,i}$ la variable aleatoria $\bar{\mathcal{P}}_{j,i}$ que representa su beneficio (normalizado) por registrar su bloque en la cadena. Para cada posición $r \in \mathcal{R}$ tenemos

$$\bar{\mathcal{P}}_{j,i}(r) = \begin{cases} \bar{\theta}_{j,i}(r), & \text{si } n_{j,i} \text{ registra su bloque en r.} \\ \\ 0, & \text{en caso contrario.} \end{cases}$$

En caso de que el valor declarado por $n_{j,i}$ para registrar su bloque en $r$ no

[1] El ganador del cluster es precisamente el jugador del cluster cuya preferencia utilizada por registrar el bloque es mayor.

pase el test de aceptación, denotamos el valor regenerado por el mecanismo por $\hat{\theta}_j(r)$ y el beneficio de $n_j$ asociado a $r$ por $\hat{P}_j(r)$.

Recordamos que asociado a cada jugador $n_{j,i} \in \mathcal{N}_j$ existe un jugador agregado formado por todos los jugadores del propio cluster excepto él mismo. Denotaremos a este jugador por $n_{-j,i}$ y usaremos los mismos subíndices para referirnos a sus preferencias (normalizadas, declaradas, utilizadas y regeneradas) y su beneficio esperado.

Del mismo modo, las preferencias normalizadas, declaradas, utilizadas y regeneradas de los ganadores de cada cluster $\mathcal{N}_j$ estarán representadas respectivamente por $\bar{\theta}_j$, $\dot{\theta}_j(r)$, $\ddot{\theta}_j(r)$ y $\hat{\theta}_j(r)$, $j = 1, ..., n$.

Por último, también existe un jugador asociado a cada $n_j \in \mathcal{N}^{(1)}$. Lo denotaremos por $n_{-j}$, y usaremos el mismo subíndice para referirnos a cualquier variable aleatoria que le haga referencia. En general, nos referiremos a $n_{-j}$ como el jugador agregado asociado a cualquier jugador $n_{j,i}$ que pertenezca a $\mathcal{N}_j$.

## 5.2. Resultados principales

De forma análoga al Capítulo 3, en esta sección presentamos primero que el mecanismo es óptimo cuando todos los jugadores son honestos. Después, demostramos diferentes resultados que nos ayudan a probar finalmente que un jugador obtiene mayor beneficio siendo honesto que siendo deshonesto.

### 5.2.1. Mecanismo óptimo para jugadores honestos

Al igual que para el mecanismo QPQ a un nivel, este nuevo mecanismo es óptimo para jugadores honestos, en el sentido de que no existe otro tal que el beneficio total de los jugadores sea mayor. Para claridad de la presentación del trabajo, dado que la demostración de este resultado se asemeja al de la Proposición 3, la omitimos.

**Proposición 8.** Si todos los jugadores son honestos, no existe ningún mecanismo $M$ distinto del mecanismo $QPQ$ a un nivel tal que, para un conjunto de posiciones $\mathcal{R}$ de la cadena, verifique

$$E[\sum_{j=1}^{n}\sum_{i=1}^{k_j}\bar{\mathcal{P}}_{j,i}^{M}] > E[\sum_{j=1}^{n}\sum_{i=1}^{k_j}\bar{\mathcal{P}}_{j,i}],$$

donde $\bar{\mathcal{P}}_{j,i}^{M}$ es la variable aleatoria que representa el beneficio (normalizado) del jugador $n_{j,i}$ cuando utiliza el mecanismo M.

### 5.2.2. Distribución de las preferencias normalizadas

Siguiendo el mismo argumento que para el mecanismo *QPQ* a un nivel, es trivial demostrar que las preferencias utilizadas de todos los jugadores ($\ddot{\theta}_{j,i}$, $1 \leqslant i \leqslant k_j$, $1 \leqslant j \leqslant n$) ya sean honestas y/o regeneradas por el mecanismo, son variables independientes entre sí e uniformemente distribuidas en (0,1).

En particular, las preferencias de los jugadores de un mismo cluster consideradas en la línea 13 de la Tabla 4.1, o las del conjunto $\mathcal{N}^{(1)}$ (línea 17 de la Tabla 4.1) son independientes y siguen la distribución uniforme normalizada.

### 5.2.3. Jugador agregado

En esta sección calculamos la distribución que siguen las preferencias del jugador agregado en un cluster $\mathcal{N}_j$ y en $\mathcal{N}^{(1)}$, respectivamente. Para ello, tomamos en consideración la Proposición 5 probada en el Capítulo 3.

Para calcular la distribución que siguen las preferencias de un jugador agregado en un cluster $\mathcal{N}_j$, usamos el mismo argumento utilizado en el Corolario 1.

**Corolario 3.** Las preferencias $\ddot{\theta}_{j,i}$ del jugador agregado $n_{-j,i}$ siguen una distribución $Beta(k_j - 1, 1)$.

*Demostración.* Por definición de jugador agregado, expresamos las preferencias utilizadas de $n_{-j,i}$ como

$$\ddot{\theta}_{-j,i} := max\{\ddot{\theta}_{j,v} \mid 1 \leqslant v \leqslant k_j,\ v \neq i\}.$$

Por hipótesis, estas variables siguen la distribución uniforme normalizada, que equivale con la $Beta(1, 1)$. Consecuentemente, por la Proposición 5 podemos afirmar que $\ddot{\theta}_{-j,i}$ sigue una distribución $Beta(k_j - 1, 1)$. □

A partir de este último resultado, es trivial deducir la distribución que siguen

las preferencias de los ganadores $n_j$ de los clusters. Solo hace falta tener en cuenta que estas pueden expresarse como sigue.

$$\ddot{\theta}_j := max\{\ddot{\theta}_{j,i} \mid 1 \leqslant i \leqslant k_j\}.$$

**Corolario 4.** Las preferencias $\ddot{\theta}_j$ de $n_j$ siguen una distribución $Beta(k_j, 1)$.

Este resultado nos ayuda a calcular la distribución que siguen las preferencias utilizadas de un jugador agregado en $\mathcal{N}^{(1)}$.

**Corolario 5.** Las preferencias $\ddot{\theta}_{-j}$ del jugador $n_{-j}$ siguen una distribución $Beta(k - k_j, 1)$.

*Demostración.* Las preferencias utilizadas de $n_{-j}$ se pueden expresar como

$$\ddot{\theta}_{-j} := max\{\ddot{\theta}_u \mid 1 \leqslant u \leqslant k_u,\ u \neq j\}.$$

Por el Corolario 4, las preferencias $\ddot{\theta}_u$ de $n_u$, $u = 1, ..., n$, siguen una distribución $Beta(k_u, 1)$. Aplicando entonces la Proposición5, podemos afirmar que $\ddot{\theta}_{-j}$ sigue una distribución $Beta(n - k_j, 1)$. □

### 5.2.4. Estrategias de los jugadores

El objetivo de esta sección es probar, al igual que para el mecanismo QPQ a un nivel, que los jugadores obtienen un mayor beneficio cuando publican sus verdaderas preferencias, independientemente de lo que haga el resto.

Es importante tener en cuenta que un jugador $n_{j,i}$ registrará su bloque en $r$ cuando su preferencia utilizada sea mayor que la de sus jugadores agregados en $\mathcal{N}_j$ y $\mathcal{N}^{(1)}$, respectivamente, es decir,

$$\ddot{\theta}_{-j,i} < \ddot{\theta}_j(r) \ \ y \ \ \ddot{\theta}_{-j} < \ddot{\theta}_j(r).$$

Calculamos a continuación el beneficio esperado de un jugador honesto.

**Proposición 9.** El beneficio esperado por $n_{j,i}$ cuando es honesto es

$$E[\bar{P}_{j,i}] = \frac{1}{k+1},$$

para cualquier estrategia del resto de jugadores

*Demostración.* Supongamos que $n_{j,i}$ es honesto. Entonces, para cualquier posición $r \in \mathcal{R}$, tenemos que la preferencia $\ddot{\theta}_{j,i}j(r)$ coincide con la verdadera preferencia normalizada $\bar{\theta}_{j,i}(r)$. El beneficio por registrar su bloque en dicha posición es entonces

$$\bar{P}_{j,i}(r) = \begin{cases} \bar{\theta}_{j,i}(r), & \text{si } \ddot{\theta}_{-j,i} < \bar{\theta}_j(r) \text{ y } \ddot{\theta}_{-j} < \bar{\theta}_j(r). \\ 0, & \text{en caso contrario.} \end{cases}$$

Asimismo, el beneficio esperado por $n_{j,i}$ es

$$E[\bar{P}_{j,i}] = \int_{\mathbb{R}} x \cdot \left[ f(x) \cdot P[\ddot{\theta}_{-j,i} < x \, , \ddot{\theta}_{-j} < x] \right] \, dx,$$

donde $f(x)$ es la función de densidad asociada a $\bar{\theta}_j$, la cual sigue la distribución uniforme normalizada.

Observamos también que $\ddot{\theta}_{-j,i}$ y $\ddot{\theta}_{-j}$ se pueden expresar como

- $\ddot{\theta}_{-j,i} = \text{máx}\{\ddot{\theta}_{j,v} \mid 1 \leqslant v \leqslant k_j, \;\; v \neq i\}$.
- $\ddot{\theta}_{-j} = \text{máx}\{\ddot{\theta}_{u,v} \mid 1 \leqslant u \leqslant n, \;\; u \neq j, \;\; 1 \leqslant v \leqslant k_j\}$.

Por la Proposición 2 y lo explicado en la Sección 5.2.2, deducimos que estas variables son independientes. Escribimos entonces

$$E[\bar{\mathcal{P}}_{j,i}] = \int_0^1 x \cdot P[\ddot{\theta}_{-j,i} < x] \cdot P[\ddot{\theta}_{-j} < x] \, dx.$$

Sabemos que las preferencias de los jugadores agregados $n_{-j,i}$ y $n_{-j}$ siguen, independientemente de la honestidad de los jugadores que lo forman, una distribución $Beta(k_j - 1, 1)$ y $Beta(k - k_j, 1)$, respectivamente (Corolarios 3 y 5). Consecuentemente, el beneficio esperado por $n_{j,i}$ cuando es honesto es, para cualquier estrategia del resto de jugadores, el siguiente:

$$E[\bar{\mathcal{P}}_{j,i}] = \int_0^1 x \cdot \left[ \int_0^x (k_j - 1) y^{k_j - 2} \, dy \right] \cdot \left[ \int_0^x (k - k_j) y^{k - k_j - 1} \, dy \right] \, dx =$$

$$= \int_0^1 x \cdot \left[ x^{k_j - 1} \right] \cdot \left[ x^{k - k_j} \right] \, dx = \int_0^1 x^k \, dx = \frac{1}{k+1}.$$

□

Observamos que el beneficio esperado por un jugador honesto es el mismo usando los mecanismos $QPQ$ a un nivel y a dos niveles (ver Proposición 6).

En cambio, el beneficio esperado por $n_{j,i}$ cambia cuando es deshonesto.

**Proposición 10.** El beneficio esperado por $n_{j,i}$ cuando es deshonesto es

$$E[\hat{P}_{j,i}] = \frac{1}{2k_j(k-k_j-1)}.$$

para cualquier estrategia del resto de jugadores.

*Demostración.* Supongamos que $n_j$ es deshonesto. Ahora, para cualquier posición $r \in \mathcal{R}$, su beneficio por registrar su bloque en dicha posición es

$$\hat{P}_{j,i}(r) = \begin{cases} \bar{\theta}_{j,i}(r), & \text{si } \ddot{\theta}_{-j,i} < \hat{\theta}_j(r) \text{ y } \ddot{\theta}_{-j} < \hat{\theta}_{j,i}(r). \\ \\ 0, & \text{en caso contrario.} \end{cases}$$

donde $\hat{\theta}_j(r)$ representa el valor regenerado (de acuerdo a la distribución uniforme normalizada) por el mecanismo $QPQ$ a dos niveles.

En este caso, el beneficio esperado por $n_j$ es

$$E[\hat{\mathcal{P}}_{j,i}] = \int_{\mathbb{R}} x \cdot \Big( f(x) \cdot P[\ddot{\theta}_{-j,i} < z, \ddot{\theta}_{-j} < z] \Big)\ dx,$$

donde $f(x)$ es la función de densidad asociada a $\bar{\theta}_j$, la cual sigue la distribución uniforme normalizada, y $z$ representa el valor regenerado del jugador, que sigue la misma distribución.

Tal y como hemos explicado en la anterior demostración, $\ddot{\theta}_{-j,i}$ y $\ddot{\theta}_{-j}$ son variables independientes que siguen, respectivamente, las distribuciones $Beta(k_j - 1, 1)$ y $Beta(k - k_j, 1)$, independientemente de la honestidad de los jugadores que lo forman. Por ello, el beneficio esperado por $n_{j,i}$ cuando es deshonesto es, para cualquier estrategia del resto de jugadores, el siguiente:

$$E[\hat{\mathcal{P}}_{j,i}] = \int_0^1 x \cdot P[\ddot{\theta}_{-j,i} < z] \cdot P[\ddot{\theta}_{-j} < z]\ dx =$$

$$= \int_0^1 x \cdot \left[ \int_0^1 \left( \int_0^z (k_j - 1) y^{k_j - 2}\ dy \right) dz \right] \left[ \int_0^1 \left( \int_0^z (k - k_j) y^{k - k_j - 1}\ dy \right) dz \right]\ dx =$$

$$= \int_0^1 x \cdot \left(\frac{1}{k_j}\right)\left(\frac{1}{k - k_j + 1}\right)\, dx =$$

$$= \frac{1}{2k_j(k - k_j + 1)}.$$

□

Comparando estas dos últimas proposiciones deducimos fácilmente que los jugadores obtienen mayor beneficio cuando son honestos que cuando son deshonestos.

**Corolario 6.** La siguiente desigualdad se cumple para cualquier jugador $n_{j,i} \in \mathcal{N}$:

$$E[\hat{P}_{j,i}] < E[\bar{P}_{j,i}].$$

*Demostración.* Suponemos que $E[\hat{P}_{j,i}] < E[\bar{P}_{j,i}]$. Por las Proposiciones 6 y 7, escribimos

$$\frac{1}{2k_j(k - k_j + 1)} < \frac{1}{k+1},$$

$$\frac{k+1}{k - k_j + 1} < 2k_j,$$

$$1 + \frac{k_j}{k - k_j + 1} < k_j + k_j.$$

Efectivamente, se cumple la igualdad pues

- Por hipótesis, todos los clusters son de tamaño mayor que uno, luego $1 < k_j$.
- Sabemos que $k > k_j$, luego $k - k_j + 1 > 1$. Consecuentemente,

$$k_j \cdot \frac{1}{k + 1 - k_j} < k_j.$$

Podemos afirmar entonces que $E[\hat{P}_{j,i}] < E[\bar{P}_{j,i}]$. □

# Capítulo 6

# Conclusiones y trabajos futuros

En primer lugar, resaltamos que hemos probado para el mecanismo $QPQ$ tanto a uno como a dos niveles, por un lado, que es óptimo para jugadores honestos (Proposiciones 3 y 8), y por otro, que independientemente de las estrategias del resto de jugadores, un jugador siempre espera un mayor beneficio normalizado cuando es honesto (Corolarios 2 y 6).

Sin embargo, no debemos olvidar que tras estos resultados hay una serie de hipótesis que no siempre se cumplen en escenarios reales, de ahí que el mecanismo $QPQ$ (a uno y a dos niveles) presenten algunas limitaciones que explicamos a continuación, y que dejamos como trabajo futuro a resolver.

(i) En el trabajo hemos supuesto que el conjunto de jugadores es fijo, es decir, que no hay jugadores que abandonen la blockchain, ni nuevos que se unan. Sin embargo, el número de usuarios de una blockchain puede cambiar con el tiempo.

(ii) También asumimos que cada vez que se subasta una posición de la cadena, todos los jugadores publican su preferencia por registrar su bloque. No obstante, puede ocurrir que en una ronda un jugador no tenga interés por registrar ningún bloque y por ello no tenga incentivo por participar.

(iii) Otra limitación podría ser que la negociación sea muy fluida y no siempre podamos contar con la información de todos los usuarios. Es decir, tengamos que aplicar el mecanismo $QPQ$ a un nivel (o a dos niveles) aunque no hayan llegado todos los mensajes. Una primera idea para solucionar este problema sería asignar valores aleatorios según una uniforme normalizada.

(iv) Hemos dado también por hecho que la comunicación es concurrente y que por lo tanto los jugadores no conocen los valores del resto a la hora de publicar sus preferencias, pero podría darse el caso de que los usuarios de la *blockchain* se comuniquen previamente entre sí, rompiendo entonces la independencia entre las preferencias. Proponemos para el análisis del problema en este caso la lectura de [6], artículo escrito por los mismos diseñadores del mecanismo *Quid Pro Quo* en el cual se estudia el problema de asignación de recursos cuando los jugadores son correlados.

Además de estas cuestiones y siguiendo la línea del trabajo, proponemos en última instancia la generalización del mecanismo $QPQ$ a $m > 2$ niveles, para posibilitar así la elección de, además del número de clusters y sus tamaños, del número de niveles para reducir el intercambio de mensajes necesarios para la ejecución del mecanismo.

# Bibliografía


[1] Santos A., Fernández Anta A., López Fernández L.; 2013. *Quid Pro Quo: A Mechanism for Fair Collaboration in Networked Systems*, PLOS ONE, Sep 5;8(9), https://doi.org/10.1371/journal.pone.0066575

[2] Natarajan H., Krause S. Karla, Gradstein H. Luskin; 2017. *Distributed Ledger Technology (DLT) and Blockchain*, FinTech note nº 1, World Bank Group, Washington D.C.

[3] Porxas N., Conejero M.; 2018 *Tecnología Blockchain: Funcionamiento, Aplicaciones y Retos Jurídicos Relacionados*, Actualidad Jurídica Uría Menéndez, revista nº 48 (2018), páginas 24-36.

[4] Vijay K. Rohatgi, A.K. Md. Ehsanes Saleh; *An introduction to probability and statistics*, 2nd edition, John Wiley, cop. 2001., Nueva York.

[5] Branislav L. Slantchev; 1939. *Game Theory: Dominance, Nash Equilibrium, Symmetry.* Department of Political Science, University of California, San Diego

[6] [6] Santos A., Fernández Anta A., Cuesta J.A., López Fernández L. (2014) *Fair Linking Mechanisms for Resource Allocation with Correlated Player Types*. In: Noubir G., Raynal M. (eds) Networked Systems. NETYS 2014. Lecture Notes in Computer Science, vol 8593. Springer, Cham