Cooperación

Sistemas Inteligentes Distribuidos

Sergio Alvarez
Javier Vázquez

Bibliografía

- *Multiagent systems: algorithmic, game-theoretic, and logical foundations* (Shoham), cap. 9, 12
- *Artificial intelligence: a modern approach* (Russell & Norvig), cap. 17
- Lamport, L. (2001). *Paxos made simple.* ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001), 51-58.

Introducción

Cooperación

¿Qué es cooperación?

- **La cooperación es un tipo de coordinación entre agentes que, en principio, no son antagónicos**
- El grado de éxito en la cooperación puede medirse por:
  - La capacidad de los agentes para mantener sus propios objetivos
  - La capacidad de permitir que otros agentes alcancen sus objetivos
- La planificación es una de las formas más sólidas de cooperación
  - Hay algunos objetivos compartidos y (quizá) un plan compartido
  - Los agentes se asignan las tareas entre ellos siguiendo el plan
- La cooperación es inherentemente compleja

Desafíos de la cooperación

- Evitar la duplicación de esfuerzos
- Evitar las interferencias o interacciones dañinas
- Evitar la sobrecarga en la comunicación
- Optimizar la sincronización del estado del mundo y de los comportamientos esperados
- Optimizar el uso de recursos computacionales

Estructuras de cooperación

- Una forma de reducir esta complejidad es mediante un **controlador centralizado**
  - Agente específico que asegura la coordinación
  - Tienen algún tipo de control sobre los objetivos de otros agentes, o sobre parte del trabajo asignado a un agente, de acuerdo con el conocimiento sobre las capacidades de cada agente
- En este caso, el objetivo final del sistema está asegurado por los objetivos del coordinador, que sustituyen los objetivos de los demás agentes del sistema
- Por ejemplo: reducir un problema MA-PDDL a un problema PDDL
- Desventajas:
  - El controlador centralizado se convierte en un cuello de botella
  - Impacto en la autonomía de los agentes
  - No es válido en sistemas multiagente donde sea necesaria la privacidad

Estructuras de cooperación

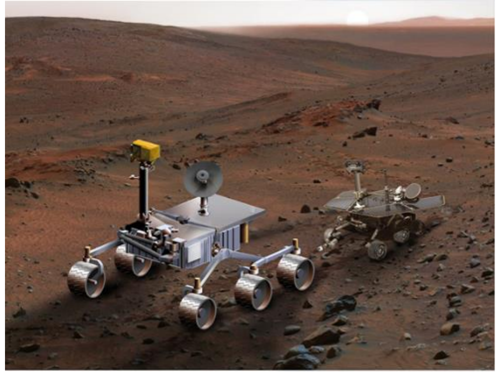
- Alternativa: distribuir el control entre todos los agentes del sistema (**control distribuido**)
- Esto implica **interiorizar el control en cada agente**
  - Capacidades de razonamiento y/o habilidades sociales
  - Razonamiento sobre las intenciones y el conocimiento de otros agentes
  - Razonamiento sobre el objetivo global del sistema y cómo resolver los conflictos que surjan
- En ámbitos donde el coste de un conflicto es alto, o la resolución del conflicto es difícil, el comportamiento completamente independiente se vuelve *no razonable* (Moses & Tennenholtz)
  - Algún tipo de estructura o modelo computacional compartido siempre es necesario

Modos de cooperación

- **Accidental:** no intencionado
- **Unilateral:** un agente ayuda intencionalmente a otro
- **Cooperación mutua:** dos o más agentes colaboran intencionalmente

Ejemplos de contextos cooperativos

- Planificación multiagente (planta de producción, reconocimiento, etc.)
- Coordinación hombre-robot (industria, cuidado médico, etc.)
- Redes de sensores (por ejemplo, seguimiento de objetivos desde múltiples puntos de vista)
- Comercio electrónico (por ejemplo, agentes web descentralizados, mercados bursátiles, redes de redes eléctricas)


Teorías y modelos

- Teorías de la cooperación
  - Resolución cooperativa de problemas (CPS)
  - Joint intentions
  - **Compromisos**
  - **Protocolos de interacción**
  - **Normas e instituciones**
  - Teamwork
- Planificación distribuida
  - PGP/GPGP
  - MA-STRIPS/MA-PDDL
  - MA-A*
- **Teoría de juegos**
  - **Coaliciones**
  - **Elección social**
  - **Diseño de mecanismos**
- Coordinación por algoritmo
  - DCOP
  - **Consenso**

Teorías y modelos

- Vamos a centrarnos en tres aspectos claves de la cooperación
  - **Teoría de coaliciones:** cómo identificar, definir e incentivar la creación de coaliciones a partir de sistemas multiagente de manera que la cooperación sea una garantía dentro de cada coalición
  - **Elección social:** cómo garantizar la cooperación en situaciones en las que los agentes tienen incentivos para cooperar, pero aun así tienen cierto grado de autonomía expresada en términos de preferencia
  - **Algoritmos de consenso:** cómo mantener la consistencia de un sistema multiagente, de manera que se llegue eficientemente a acuerdos incluso en situaciones de fallo o (hasta cierto punto) presencia de agentes maliciosos

Teorías y modelos

- Más adelante también veremos:
  - **Diseño de mecanismos:** le daremos la vuelta a la teoría de juegos para diseñar juegos donde los agentes se vean incentivados a cooperar, voluntaria o involuntariamente, en beneficio de los objetivos del sistema
  - **Enfoques simbólicos:** uso de los símbolos (lenguaje) para crear protocolos, generar compromisos y aplicar normas para tratar el sistema multiagente como una institución o una organización

Coaliciones

Cooperación

Teoría de Coaliciones

- Las coaliciones emergen a partir de una identificación mutua de objetivos comunes o similares bajo el supuesto de que la coalición será capaz de cumplir estos objetivos más eficientemente que de manera individual
  - Partidos políticos
  - Equipos deportivos
  - Relaciones personales: grupos de amigos, matrimonios
  - Asociaciones (de vecinos, de estudiantes, de empresas, …)
  - Grupos de prácticas

- La **Teoría de Coaliciones** es la parte de la Teoría de Juegos que estudia **cómo se pueden formar las coaliciones y cuál es su comportamiento estratégico desde el punto de vista colectivo** (no individual)
- Suponemos que, **formada la coalición, hay un incentivo para que los agentes, racionalmente, cooperen**
- Cooperación es coordinación: **la contribución de cada agente será relevante**

Suposiciones previas

- En Teoría de Coaliciones, se asumen como ciertas siempre las siguientes propiedades sobre el sistema multiagente:
  - Las recompensas son redistribuibles: transferibles entre los miembros de una coalición
  - No existen diferentes tipos de recompensa (e.g. no hay *divisas*)
  - Las recompensas se definen sobre la coalición, en lugar de sobre los individuos
- Encontrar una coalición es, bajo estos supuestos, reducible a la **búsqueda de la coalición que maximice las recompensas, la colectiva y las individuales, recibidas**

Definición

- Un **juego coalicional con utilidad transferible** es una tupla $\langle N, v \rangle$, donde:
  - $N = \{i, j, k \dots\}$ es un conjunto finito de agentes
  - $v: 2^N \rightarrow \mathbb{R}$ es una función que retorna, para cada coalición $S \subseteq N$, una recompensa $v(S)$ tal que $v(\emptyset) = 0$
- Los miembros de una coalición S pueden redistribuirse su recompensa $v(S)$, es decir, pueden realizar cualquier posible asignación de esa recompensa entre los diferentes miembros
- A partir de esta definición, vamos a responder a las siguientes preguntas:
  - ¿Qué coalición se debería formar?
  - ¿Cómo debería la coalición repartir la recompensa entre sus miembros?

Juegos superaditivos

- ¿En qué situaciones la coalición que se forme será la **gran coalición** (N)?
- Se dice de un juego $G = \langle N, v \rangle$ que es **superaditivo** cuando cualquier posible par de coaliciones que se puedan extraer de N puede coexistir sin interferirse mutuamente:

$$\forall S, T \subset N, [S \cap T = \emptyset \rightarrow v(S \cup T) \geq v(S) + v(T)]$$

- En un juego superaditivo, N recibe la máxima recompensa posible
- En adelante, vamos a suponer que estamos en un juego superaditivo
  - Existen métodos para particionar un juego no superaditivo en diversos juegos superaditivos

Reparto de recompensas

- Una función de reparto $\psi_i(N, v)$ asigna un valor de recompensa para cada agente i de una coalición N con recompensa v
  - Buscaremos cumplir dos propiedades al diseñar un reparto: equidad y estabilidad
- **Equidad:** ha de ser *justo*, e.g.
  - **Valor de Shapley:** los miembros deberían recibir una recompensa proporcional a su contribución
  - Reparto igualitario: cada posible subcoalición recibe como mínimo la recompensa que podría recibir de manera independiente
- **Estabilidad:** una vez repartida la recompensa, ningún agente está incentivado a abandonar la coalición, e.g.
  - **El Núcleo (*Core*):** un reparto es justo si no hay ninguna subcoalición que podría conseguir una mejor recompensa siendo independiente
  - Nucleolo: un reparto es justo cuando, entre todas las subcoaliciones posibles, la mayor insatisfacción (diferencia entre lo que la subcoalición recibe y lo que recibiría de manera independiente) es lo más pequeña posible

Valor de Shapley

- Cada miembro de una coalición debería recibir una recompensa individual **proporcional a su contribución marginal**
  - Contribución marginal: contribución neta de un miembro cuando se incorpora a un conjunto de agentes
- Caso problemático:
  - Si todos los agentes son imprescindibles: $S \neq N \land v(N) = 1 \land v(S) = 0$ o equivalentemente $\forall i$: $v(N) - v(N \backslash \{i\}) = 1$.
  - **¿Cómo distribuimos en este caso?**
- **Necesitamos un sistema de reparto que permita otorgar pesos a cada agente**
  - De esta manera, podríamos repartir peso entre agentes con la misma contribución

Axioma: Eficiencia

- La suma de las recompensas individuales para todos los agentes de una coalición debería coincidir con la recompensa de esa coalición:

$$\sum_{i \in N} \psi_i(N, v) = v(N)$$

Axioma: Simetría

- Los agentes i y j se consideran **intercambiables** con respecto a v si **siempre contribuyen lo mismo, sea cual sea la coalición en la que estén**:

$$\forall S \subseteq N \backslash \{i, j\}: v(S \cup \{i\}) = v(S \cup \{j\})$$

- Si dos agentes i y j son intercambiables, entonces:

$$\psi_i(N, v) = \psi_j(N, v)$$

es decir, deberían recibir la misma recompensa individual

Axioma: *Dummy player*

- Un agente i se considera ***dummy player*** si su contribución neta en cualquier coalición posible es 0:

$$\forall S \subseteq N: v(S \cup \{i\}) = v(S)$$

- Si un agente i es *dummy player:*

$$\psi_i(N, v) = 0$$

es decir, no debería recibir nada

Axioma: Aditividad

- Si podemos separar un juego en dos partes con sus recompensas (por ejemplo, podemos hacer una división en una secuencia de tareas), entonces deberíamos poder también descomponer el reparto de recompensas:

$$\forall i, \forall v_1, v_2:$$

$$(v_1 + v_2)(S) = v_1(S) + v_2(S)$$

$$\rightarrow$$

$$\psi_i(N, v_1 + v_2) = \psi_i(N, v_1) + \psi_i(N, v_2)$$

Función de valor de Shapley

- Antes de ver la formalización, definamos el **valor de Shapley** de manera informal: la idea principal es calcular la contribución marginal esperada
- Para hacerlo de manera justa, hemos de considerar TODAS las posibles maneras, ordenadas, de construir una coalición
- Lo haremos sumando las contribuciones marginales resultantes de añadir el agente i a todas las posibles subcoaliciones ordenadas dentro de N
  - Ponderadas por el número de coaliciones que comienzan por esa subcoalición
  - Al final tendremos que dividir entre todas las posibles subcoaliciones

Función de valor de Shapley

- Ejemplo: $N = \{a, b, c\}$
- Si queremos saber la contribución marginal esperada de a, necesitaremos considerar los casos:
  - $v_1 = v(\{a\}) - v(\emptyset)$ (ponderada por 2 secuencias que comienzan por a: a, b, c y a, c, b)
  - $v_2 = v(\{b, a\}) - v(\{b\})$ (sólo 1 secuencia empieza por b, a: b, a, c)
  - $v_3 = v(\{c, a\}) - v(\{c\})$ (sólo 1 secuencia empieza por c, a: c, a, b)
  - $v_2 = v(\{b, c, a\}) - v(\{b, c\})$ (hay 2 maneras de ordenar $\{b, c\}$: b, c, a y c, b, a)
- En total, 6 secuencias posibles. El valor de Shapley será

$$\frac{1}{6} \cdot [2 \cdot v_1 + 1 \cdot v_2 + 1 \cdot v_3 + 2 \cdot v_4]$$

Función de valor de Shapley

- Dado un juego coalicional $\langle N, v \rangle$, el **valor de Shapley** para el reparto se define, para todo $i \in N$, como:

$$\phi_i(N,v) = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} |S|! \, \underbrace{(|N| - |S| - 1)!}_{\text{cuántas secuencias comienzan por } S \cup \{i\}} \, \underbrace{[v(S \cup \{i\}) - v(S)]}_{\text{contribución marginal de } i \text{ cuando se añade a } S}$$

dividimos por el número total de ordenaciones

sumatorio para todos los subconjuntos de N que no tienen i

posibles maneras ordenadas de construir S

Función de valor de Shapley

- Es demostrable que existe un único reparto $\phi(N, v)$ que distribuye la recompensa v de manera que se cumplen los cuatro axiomas: Eficiencia, Simetría, *Dummy player* y Aditividad
- Este único reparto coincide con el resultado de aplicar la función de valor de Shapley
- Eficiencia: $v(\{a\}) = 2, v(\{b\}) = 5, v(\{a, b\}) = 10$

$$\left.\begin{aligned}
\phi_a(N, v) &= \frac{1}{2} \cdot \left[1 \cdot \left(v(\{a\}) - v(\emptyset)\right) + 1 \cdot \left(v(\{a, b\}) - v(\{b\})\right)\right] = \frac{1}{2} \cdot [2 + 5] = 3.5 \\
\phi_b(N, v) &= \frac{1}{2} \cdot \left[1 \cdot \left(v(\{b\}) - v(\emptyset)\right) + 1 \cdot \left(v(\{a, b\}) - v(\{a\})\right)\right] = \frac{1}{2} \cdot [5 + 8] = 6.5
\end{aligned}\right\} = \mathbf{10}$$

Función de valor de Shapley

- Simetría: $v(\{a\}) = 3, v(\{b\}) = 3, v(\{a, b\}) = 10$

$$\phi_a(N, v) = \frac{1}{2} \cdot \left[1 \cdot \left(v(\{a\}) - v(\emptyset)\right) + 1 \cdot \left(v(\{a, b\}) - v(\{b\})\right)\right] = \frac{1}{2} \cdot [3 + 7] = 5$$

$$\phi_b(N, v) = \frac{1}{2} \cdot \left[1 \cdot \left(v(\{b\}) - v(\emptyset)\right) + 1 \cdot \left(v(\{a, b\}) - v(\{a\})\right)\right] = \frac{1}{2} \cdot [3 + 7] = 5$$

- *Dummy player*: $v(\{a\}) = 0, v(\{b\}) = 10, v(\{a, b\}) = 10$

$$\phi_a(N, v) = \frac{1}{2} \cdot \left[1 \cdot \left(v(\{a\}) - v(\emptyset)\right) + 1 \cdot \left(v(\{a, b\}) - v(\{b\})\right)\right] = \frac{1}{2} \cdot [0 + 0] = 0$$

$$\phi_b(N, v) = \frac{1}{2} \cdot \left[1 \cdot \left(v(\{b\}) - v(\emptyset)\right) + 1 \cdot \left(v(\{a, b\}) - v(\{a\})\right)\right] = \frac{1}{2} \cdot [10 + 10] = 10$$

Función de valor de Shapley

- Aditividad:

$$v_1(\{a\}) = 3, v_1(\{b\}) = 6, v_1(\{a,b\}) = 10$$

$$v_2(\{a\}) = 5, v_2(\{b\}) = 4, v_2(\{a,b\}) = 10$$

$$v(\{a\}) = 8, v(\{b\}) = 10, v(\{a,b\}) = 20$$

$$\phi_a(N, v_1) = \frac{1}{2} \cdot \left[1 \cdot \left(v_1(\{a\}) - v_1(\emptyset)\right) + 1 \cdot \left(v_1(\{a,b\}) - v_1(\{b\})\right)\right] = \frac{1}{2} \cdot [3 + 4] = 3.5$$

$$\phi_b(N, v_1) = \frac{1}{2} \cdot \left[1 \cdot \left(v_1(\{b\}) - v_1(\emptyset)\right) + 1 \cdot \left(v_1(\{a,b\}) - v_1(\{a\})\right)\right] = \frac{1}{2} \cdot [6 + 7] = 6.5$$

$$\phi_a(N, v_2) = \frac{1}{2} \cdot \left[1 \cdot \left(v_2(\{a\}) - v_2(\emptyset)\right) + 1 \cdot \left(v_2(\{a,b\}) - v_2(\{b\})\right)\right] = \frac{1}{2} \cdot [5 + 6] = 5.5$$

$$\phi_b(N, v_1) = \frac{1}{2} \cdot \left[1 \cdot \left(v_2(\{b\}) - v_2(\emptyset)\right) + 1 \cdot \left(v_2(\{a,b\}) - v_2(\{a\})\right)\right] = \frac{1}{2} \cdot [4 + 5] = 4.5$$

$$\boldsymbol{\phi_a(N, v) = \frac{1}{2} \cdot \left[1 \cdot \left(v(\{a\}) - v(\emptyset)\right) + 1 \cdot \left(v(\{a,b\}) - v(\{b\})\right)\right] = \frac{1}{2} \cdot [8 + 10] = 9}$$

$$\boldsymbol{\phi_b(N, v) = \frac{1}{2} \cdot \left[1 \cdot \left(v(\{b\}) - v(\emptyset)\right) + 1 \cdot \left(v(\{a,b\}) - v(\{a\})\right)\right] = \frac{1}{2} \cdot [10 + 12] = 11}$$

Equidad y estabilidad

- La función de valor de Shapley propone una interpretación de justicia para repartir la recompensa colectiva entre los miembros de una coalición de forma equitativa
  - Esta función no sólo tiene aplicación en Teoría de Juegos: también en explicabilidad en aprendizaje automático
- Garantizar equidad, sin embargo, no garantiza automáticamente estabilidad
  - Dado un reparto justo, ¿qué previene a los agentes buscar formar coaliciones más pequeñas donde su recompensa y/o su reparto justo sean más provechosos?

Ejemplo: *voting game* (Leyton-Brown, Shoham)

- Un parlamento se compone de cuatro partidos políticos A, B, C, D que tienen, respectivamente, $45, 25, 15$ y 15 diputados
- Para una determinada medida política, hay un presupuesto de 100 millones, para el cual tienen que decidir qué parte de ese presupuesto estará bajo el control de cada partido
- Es necesario un voto mayoritario (>50%) para aprobar el presupuesto
- Si no hay acuerdo, el presupuesto no se ejecutará y por lo tanto tampoco se repartirá
- Valores de Shapley: $50, 16.67, 16.67, 16.67$
- **¿Hay incentivos para que alguna subcoalición decida partir la gran coalición?**

Ejemplo: *voting game* (Leyton-Brown, Shoham)

- Un parlamento se compone de cuatro partidos políticos A, B, C, D que tienen, respectivamente, $45, 25, 15$ y 15 diputados
- Valores de Shapley: $50, 16.67, 16.67, 16.67$
- ¿Hay incentivos para que alguna subcoalición decida partir la gran coalición?
- Cualquier coalición entre B, C o D a solas con A ofrecerá una mayor recompensa individual
  - Por ejemplo, D podría negociar con A una coalición (45+25 = 70 > 50% de los votos) con un reparto de 75, 25: **inestabilidad**

El Núcleo (*The Core*)

- Una forma de medir si la estabilidad es posible es encontrando el conjunto de repartos posibles bajo los que todos los agentes siempre querrían formar la gran coalición: **el núcleo**
- Se dice que un vector de recompensas x está en el **núcleo** de un juego coalicional $\langle N, v\rangle$ si, y sólo si:

$$\forall S \subseteq N, \sum_{i \in S} x_i \geq v(S)$$

- Es decir: para cualquier subcoalición posible, la suma de recompensas de x para sus agentes es por lo menos igual de grande que la que conseguiría colectivamente la subcoalición

Volviendo al *voting game*…

- Un parlamento se compone de cuatro partidos políticos A, B, C, D que tienen, respectivamente, $45, 25, 15$ y 15 diputados
- Valores de Shapley para A, B, C, D: $50, 16.67, 16.67, 16.67$
- El conjunto de subcoaliciones que pueden conseguir una mayoría son:

$$\{A, B\}, \{A, C\}, \{A, D\}, \{B, C, D\}, \{A, B, C, D\}$$

- La subcoalición $\{B, C, D\}$ forma mayoría, por lo que $v(\{B, C, D\}) = 100$
- Cualquier reparto x que sume menos de 100 para $\{B, C, D\}$ representará un incentivo para que esta subcoalición se separe de la gran coalición
- Sin embargo, cualquier reparto x que deje al agente A con una recompensa de 0 representará un incentivo para que A negocie formar una coalición con B, C o D
- En este juego **no hay núcleo**

Voting game con otra mayoría

- Un parlamento se compone de cuatro partidos políticos A, B, C, D que tienen, respectivamente, $45, 25, 15$ y 15 diputados
- Valores de Shapley para A, B, C, D: $50, 16.67, 16.67, 16.67$
- Cambiamos la mayoría para que ahora el mínimo necesario sea el 80%
- En este caso, sólo tres coaliciones pueden formar un acuerdo válido: $\{A, B, C\}$, $\{A, B, D\}$ y $\{A, B, C, D\}$
  - Todos los acuerdos válidos pasan por A o B (aunque $v(\{A, B\}) = 0$)
  - Cualquier reparto completo de los 100 millones entre A y B pertenece al **núcleo**

Propiedades del núcleo

- No hay garantías de existencia de núcleo en un juego coalicional en el caso general
- Puede existir más de un vector de recompensas x en el núcleo

Juegos simples

- El *voting game* pertenece a un tipo especial de juegos coalicionales llamados juegos simples
- Un juego $G = \langle N, v \rangle$ es simple si la recompensa colectiva es booleana:

$$\forall S \subset N : v(S) \in \{0,1\}$$

- *Voting game* es un juego simple porque o bien hay acuerdo (una mayoría) o bien no lo hay
  - El reparto x de las recompensas individuales se puede tratar como un proceso aparte

- Además, hemos podido observar que pueden existir agentes que son decisivos para que la coalición se pueda formar
- Un agente i es un agente con veto si $v(N \setminus \{i\}) = 0$
- **En un juego simple, el núcleo está vacío si, y sólo si, no hay agentes con veto**
- **Si hay agentes con veto, el núcleo está formado por todos los vectores x posibles en los que los agentes sin veto reciben 0**

Juegos convexos

- Se dice de un juego $G = \langle N, v \rangle$ que es convexo cuando dos subcoaliciones obtienen una recompensa igual o mayor que por separado tras descontar cualquier superposición:

$$\forall S, T \subset N \colon v(S \cup T) \geq v(S) + v(T) \ - v(S \cap T)$$

- **Un juego convexo tiene un núcleo no vacío**
- **En un juego convexo, el valor de Shapley para su coalición pertenece al núcleo**

Ejemplo de juego no convexo: *voting game*

- Un parlamento se compone de cuatro partidos políticos A, B, C, D que tienen, respectivamente, $45, 25, 15$ y 15 diputados
- Valores de Shapley para A, B, C, D: $50, 16.67, 16.67, 16.67$
- El conjunto de subcoaliciones que pueden conseguir una mayoría son:

$$v(N) = v(\{A, B\}) = v(\{A, C\}) = v(\{A, D\}) = v(\{A, B, C\}) = v(\{A, B, D\})$$
$$= v(\{B, C, D\}) = 1$$

$$v(\{A\}) = v(\{B\}) = v(\{C\}) = v(\{D\}) = v(\{B, C\}) = v(\{B, D\}) = v(\{C, D\}) = 0$$

- Un caso: ¿$v(\{A, B\} \cup \{A, C\}) \geq v(\{A, B\}) + v(\{A, C\}) - v(\{A\})$?
  - $1 \geq 1 + 1 - 0$?
  - **No, por lo tanto, este juego no es convexo**

Ejemplo de juego convexo: *airport game*

- Varias ciudades cercanas necesitan un aeropuerto
- Si acuerdan construir un solo aeropuerto regional, las ciudades tendrán que compartir el coste del aeropuerto, que dependerá del tamaño del avión más grande requerido entre las ciudades
- Si no hay acuerdo, cada ciudad tendrá que construir su propio aeropuerto
- N es el conjunto de ciudades y, $\forall S \subseteq N$, $v(S)$ es el ahorro que consigue la coalición, que es igual a la suma de los costes de construir un aeropuerto para cada ciudad en S menos el coste de construir un aeropuerto para la pista más grande requerida por cualquier ciudad en S:

$$v(\{A,C\} \cup \{B,C\}) \geq v(\{A,C\}) + v(\{B,C\}) \ - v(\{C\})$$

Formación de coaliciones

- Hemos visto métodos que permiten evaluar la equidad y la estabilidad de una coalición
  - Son los dos métodos más populares, aunque hay alternativas
- El subcampo de la Teoría de Formación de Coaliciones se ocupa de estudiar cómo se puede dividir un grupo de agentes en (potencialmente) múltiples coaliciones
  - Los métodos que hemos visto permiten evaluar las coaliciones, antes de formarse o una vez formadas
  - Sin embargo, este subcampo es muy amplio e incluye el estudio de técnicas de negociación y acuerdo (equilibrios, *blocking*, *bargaining*, …) para la búsqueda de coaliciones candidatas
- Suponiendo que una coalición está formada, continuamos con Teoría de la Elección Social: cómo llegar a acuerdos cuando, además de compartir objetivos comunes, los agentes tienen preferencias individuales

Elección social

Cooperación

Elección social: definición

- Teoría de la elección social
  - ¿Cómo acordar una decisión cuando hay agentes con incentivos asimétricos que aceptan cooperar?
  - El resultado de un mecanismo (o función) de elección social es una agregación de las preferencias de todos los agentes
- Dado un conjunto de agentes $N = \{i, j, \dots\}$, una función de elección social es una función $f: V \rightarrow O$ tal que
  - O es un conjunto de resultados (*outcomes*), e.g.: recompensas para cada agente, asignaciones de tareas, decisiones booleanas, elección de líder, …
  - $V = \{v_i, v_j, \dots\}$ es un conjunto de preferencias $o_x \succ o_y$ para cada agente:
    - La relación de preorden $\succ$ es total: $\forall x, y$: $(o_x \succ o_y \vee o_y \succ o_x) \wedge \neg(o_x \succ o_y \wedge o_y \succ o_x)$
    - $\succ$ es transitiva: $o_x \succ o_y \wedge o_y \succ o_z \rightarrow o_x \succ o_z$

Ejemplo de elección social

- Imaginemos que tenemos que los agentes 1, 2 y 3 (N) tienen que elegir un nuevo líder (O) entre A, B o C, a partir de esta tabla de preferencias (V):

1	2	3
A B C	B C A	C A B

→ ***¿A, B o C?***

- ¿Qué función $f: V \to O$ nos podría resolver la elección?

Paradoja de Condorcet

1	2	3
A B C	B C A	C A B

- Una función de elección social habitual es la mayoría simple: $\frac{N}{2} + 1$
- Como hay empates opción contra opción, vamos a intentar aplicar la mayoría sobre los órdenes de preferencia
- Podemos observar, por ejemplo, que se elige A sobre B dos veces sobre tres, y también se elige B sobre C dos veces sobre tres
- Implica eso que podemos suponer que, colectivamente, ¿se elige A sobre C?
- ¡No! **Las diferentes mayorías pueden estar compuestas por diferentes agentes**
  - La mayoría que compone la preferencia colectiva “A sobre B” es diferente de la mayoría que compone la preferencia colectiva “B sobre C”

Paradoja de Condorcet

- **La transitividad en las preferencias individuales no implica transitividad en las preferencias sociales**
- Usar reglas sobre mayorías como funciones de elección social crea contradicciones cuando hay más de dos conjuntos de preferencias

Elección social: propiedades deseables

- **Siempre un ganador**
  - Siempre hay al menos una opción ganadora
- **Criterio de Condorcet**
  - Para cada alternativa y a la ganadora x, la mayoría siempre prefiere x sobre y
- **Criterio de Pareto** (Pareto-eficiencia)
  - Si todos los agentes prefieren x sobre y, y nunca puede ser la opción ganadora
- **Utilidad** (no dictatorial)
  - Por lo menos dos agentes pueden influenciar el resultado de la elección
- **Monotonicidad**
  - Si y es la opción ganadora para un conjunto específico V, para cualquier conjunto V' cuyo único cambio con respecto a V es que un agente clasifica y todavía más alto, la opción y debe seguir siendo la ganadora
- **Independencia de las alternativas irrelevantes**
  - Si x es la opción ganadora e y no lo es, si los agentes cambian sus preferencias sobre cualquier alternativa, pero no sobre la preferencia entre x e y, y no debería nunca convertirse en la ganadora

Teorema de la imposibilidad de Arrow:
es imposible tener una función de elección social Pareto-eficiente y no dictatorial que no viole la independencia de la alternativas irrelevantes

Algunas funciones de elección social

- **Regla de la mayoría**
  - La opción elegida por lo menos $\frac{|N|}{2} + 1$ veces en primer puesto es la ganadora
- **Método de Condorcet**
  - x está entre las opciones ganadoras si, para cada alternativa y, la mayoría prefiere x sobre y
- **Método de pluralidad**
  - Las opciones ganadoras son las que tienen la mayor cantidad de primeros puestos
- **Método Borda**
  - Cada opción obtiene $|N| - r$ puntos de cada agente i, siendo r el rango de la opción en V_i
- **Liebre**
  - Se elimina la opción peor clasificada y se repite la votación hasta que solo queda una opción
- **Votación secuencial por pares con una agenda fija**
  - Previamente a la elección se elige un orden, y se vota entre la primera y la segunda, luego se vota entre la opción previamente ganadora y la tercera, etc.
- **Dictadura**
  - Se decide que un agente es dictatorial y la preferencia de este agente será la elección social

Consenso

Cooperación

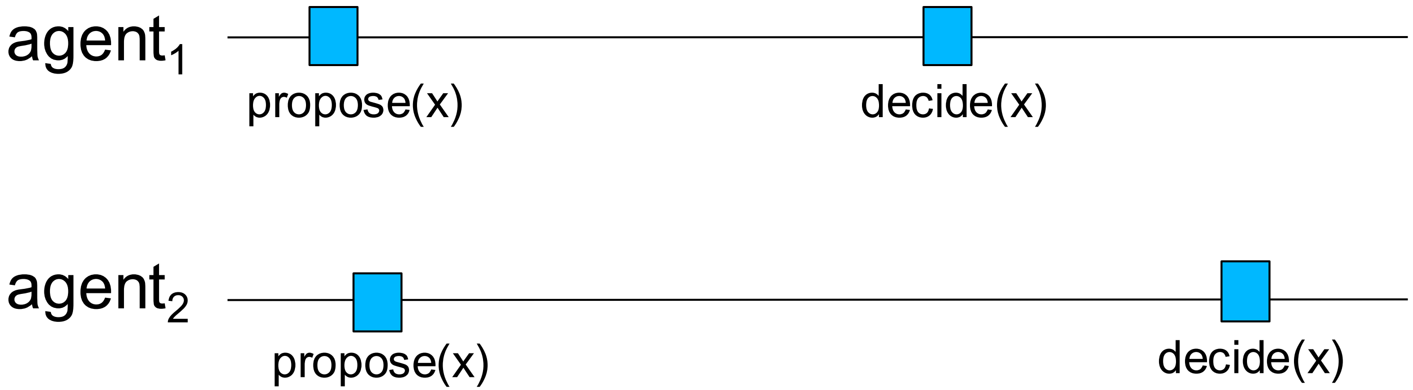
Algoritmos de consenso distribuido

- El problema del consenso: cómo lograr que el sistema multiagente sea fiable y coherente en presencia de agentes *defectuosos* (maliciosos o inestables)
- Un protocolo de consenso es un protocolo de interacción que es tolerante a un número (limitado) de agentes defectuosos
- En general, un protocolo de consenso describe cómo mantener un valor común, único y global, en un sistema multiagente, a través de la elección del líder (que decidirá el valor) o una votación por mayoría
  - Un protocolo de consenso define los procesos que cada agente tiene que ejecutar para leer y/o actualizar este valor
- Este valor único puede ser muy complejo, por ejemplo, el *ledger* de Bitcoin

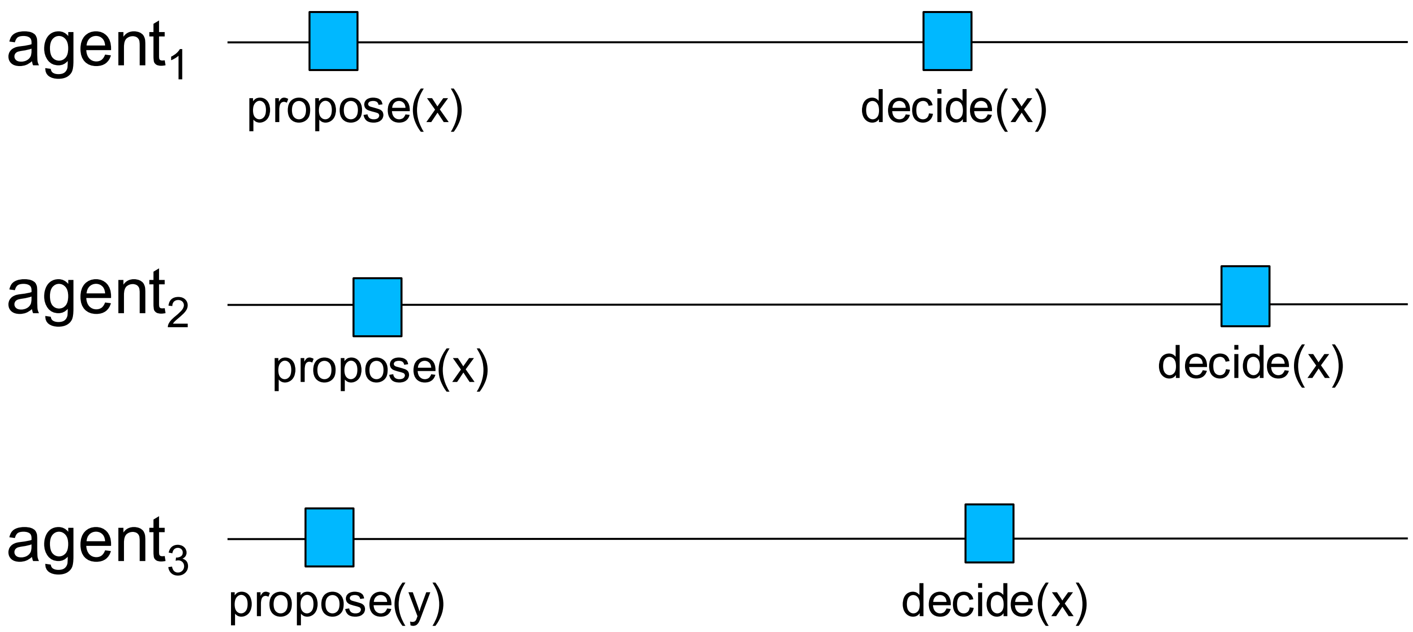
Propiedades deseables

- Un algoritmo de consenso distingue entre propuestas, decisiones y acuerdos: se llega a un acuerdo final cuando hay una mayoría que decide el mismo valor
- Se dice que un algoritmo de consenso es tolerante a fallos cuando cumple las siguientes propiedades:
  - **Terminación**
    - En tiempo finito, cada proceso (= agente) correcto decide algún valor
  - **Integridad (Validez)**
    - Un nodo decide como máximo una vez
    - Cualquier valor decidido es un valor propuesto
    - Si todos los procesos correctos propusieron el mismo valor, entonces cualquier proceso correcto debe decidir ese valor
  - **Acuerdo**
    - Todo proceso correcto debe coincidir en el mismo valor

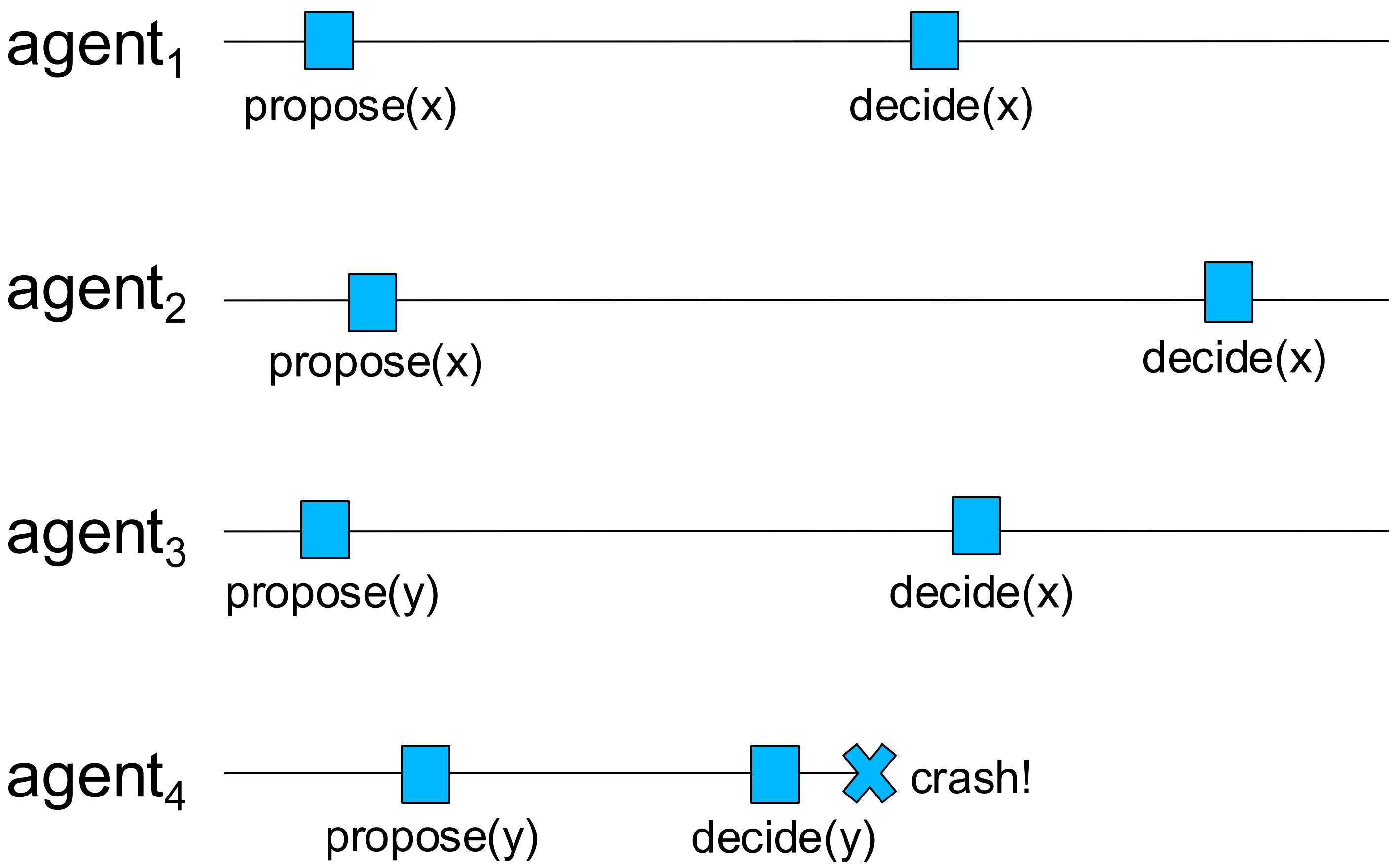
¿Es esto consenso?



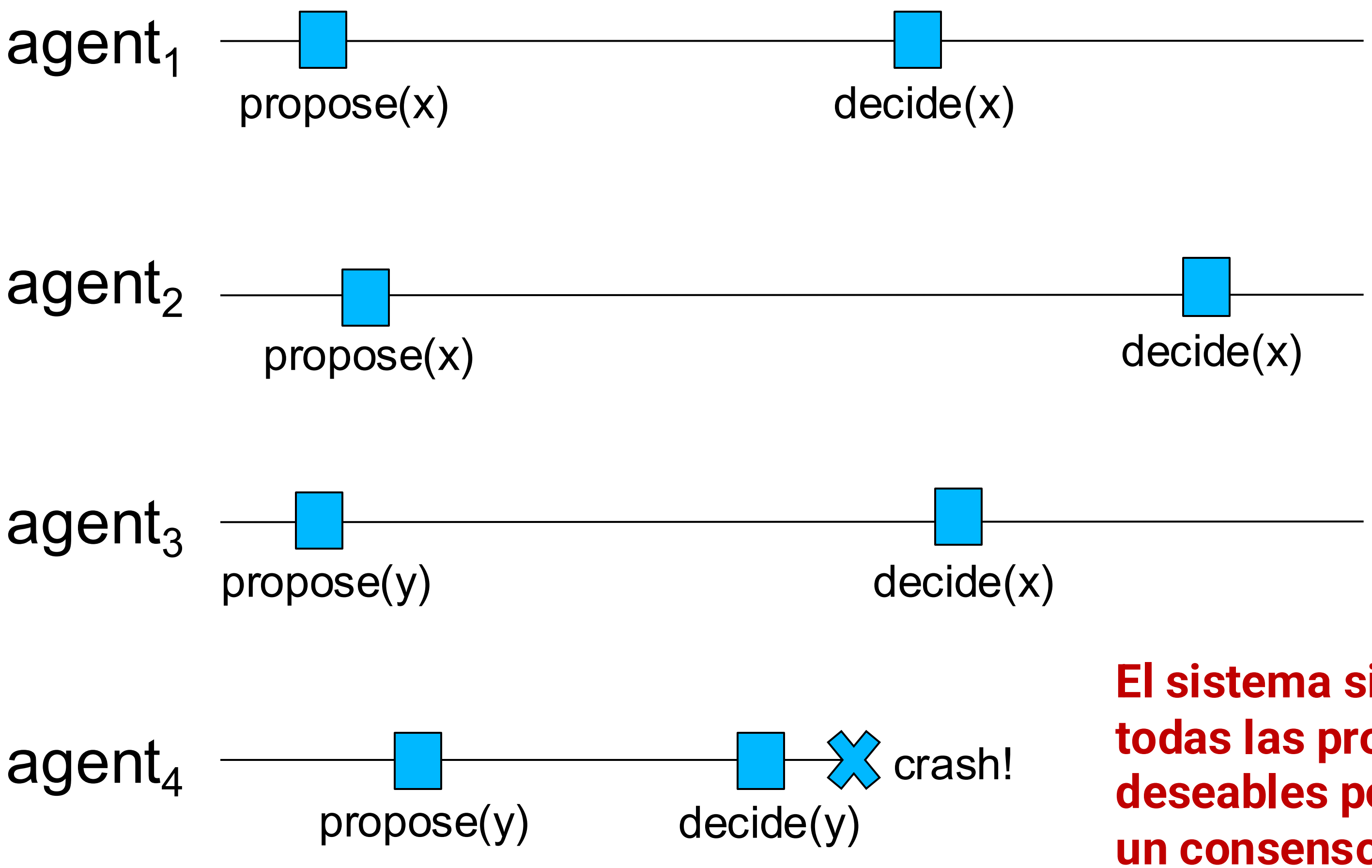
¿Es esto consenso?



¿Es esto consenso?



¿Es esto consenso?

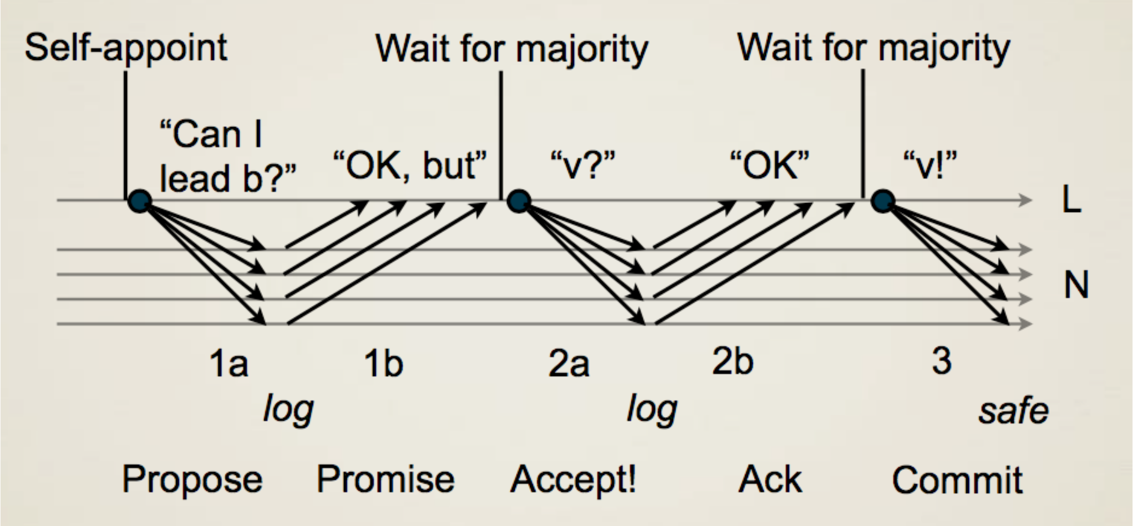


El sistema sigue cumpliendo todas las propiedades deseables por lo que esto es un consenso

Paxos

Paxos

- Paxos (Lamport, Malkhi, Zhou, 2010) es actualmente la familia de algoritmos de consenso distribuido más popular
  - Raft es una propuesta similar, (supuestamente) más simple formalmente
- Una máquina de estado se replica en todos los agentes del sistema



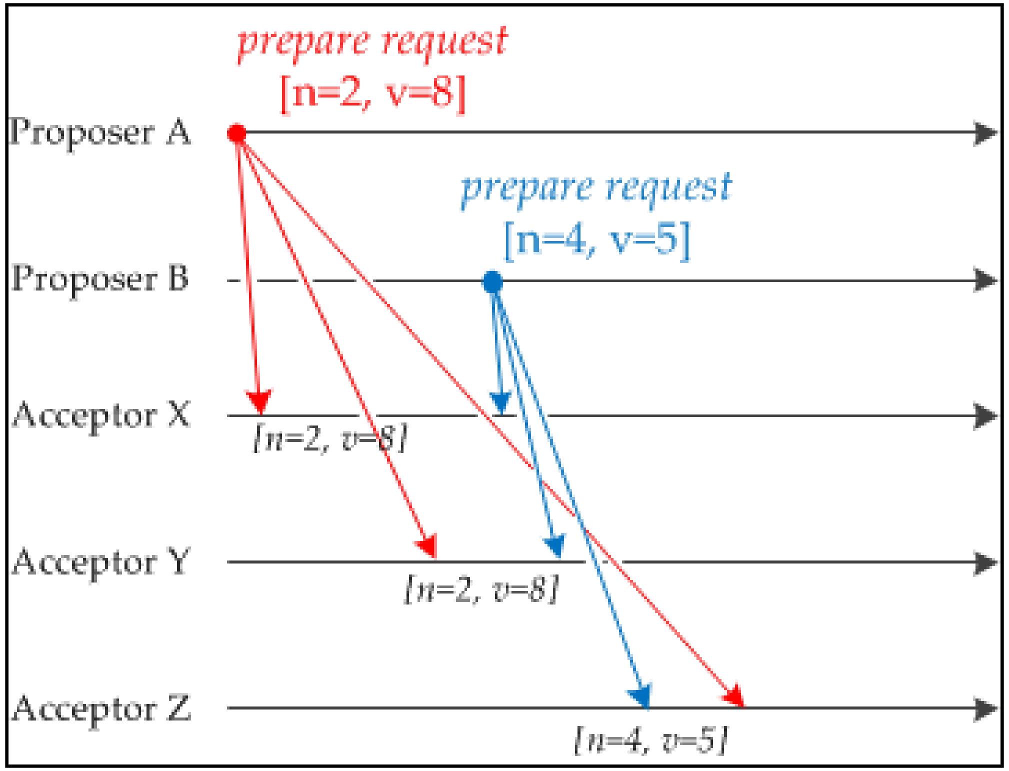
Paxos

- En Paxos, hay tres roles de agente:
  - Proponente
  - Aceptador
  - Aprendiz
- Cualquier agente puede convertirse en proponente en cualquier momento
- Un proponente crea un quórum de aceptantes para su propuesta
- Un quórum es un subconjunto del conjunto de agentes que es mayoritario, es decir, su tamaño es mayor que la mitad del tamaño del conjunto completo
  - $\{A, C, D\}$ es un quórum posible para $\{A, B, C, D\}$
  - El resto de los agentes son aprendices para el alcance de la propuesta

Fase 1a: Preparar

- Un proponente P crea una propuesta identificada con un número N adjuntando su valor propuesto v
- N debe ser mayor que cualquier número de propuesta anterior utilizado o recibido por el proponente P
- P envía un mensaje de preparación, que contiene esta propuesta, a un quórum de aceptador
- El proponente P decide quién está en el quórum

Fase 1a: Preparar

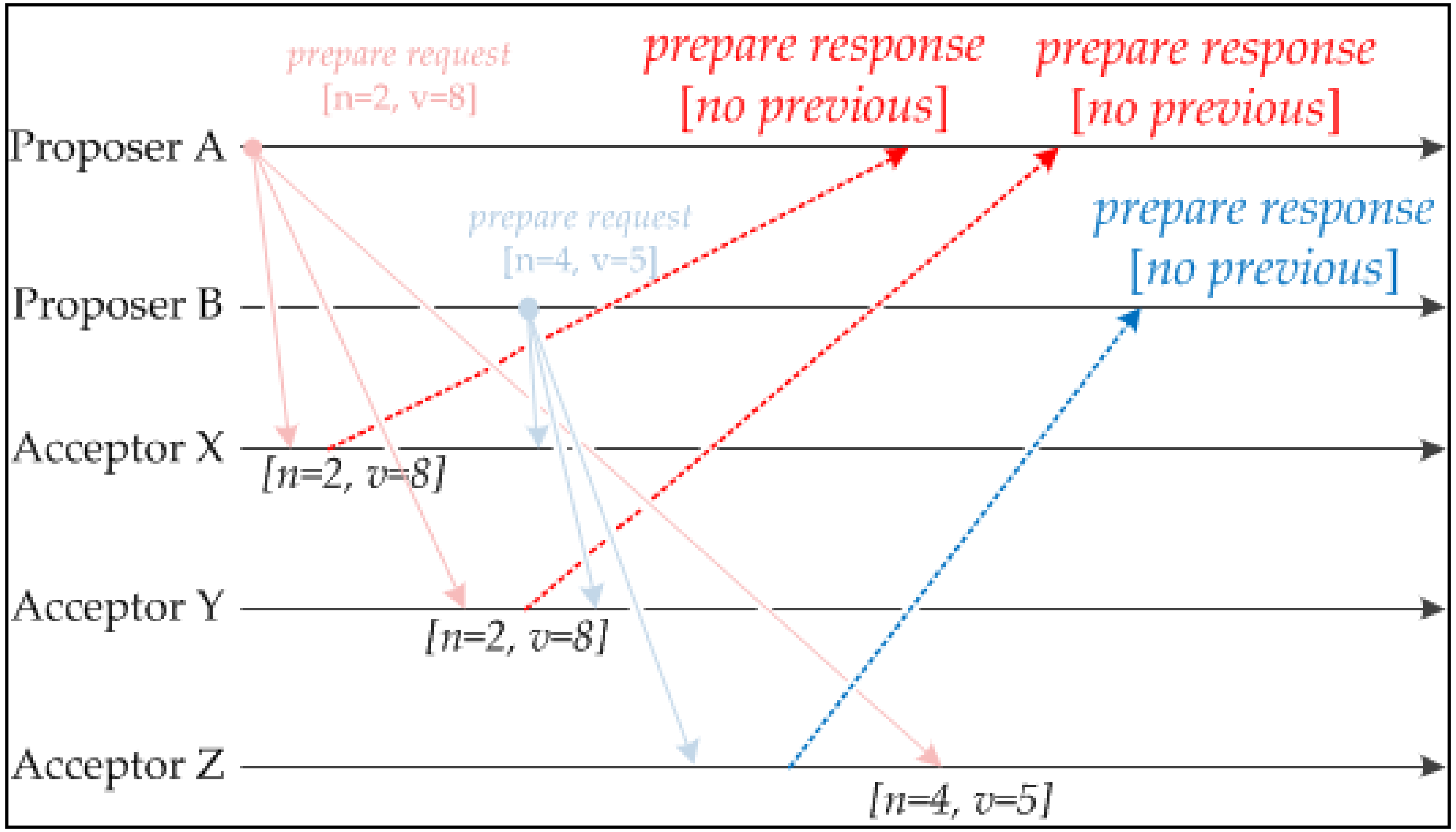


https://medium.com/@angusmacdonald/paxos-by-example-66d934e18522

Fase 1b: Prometer

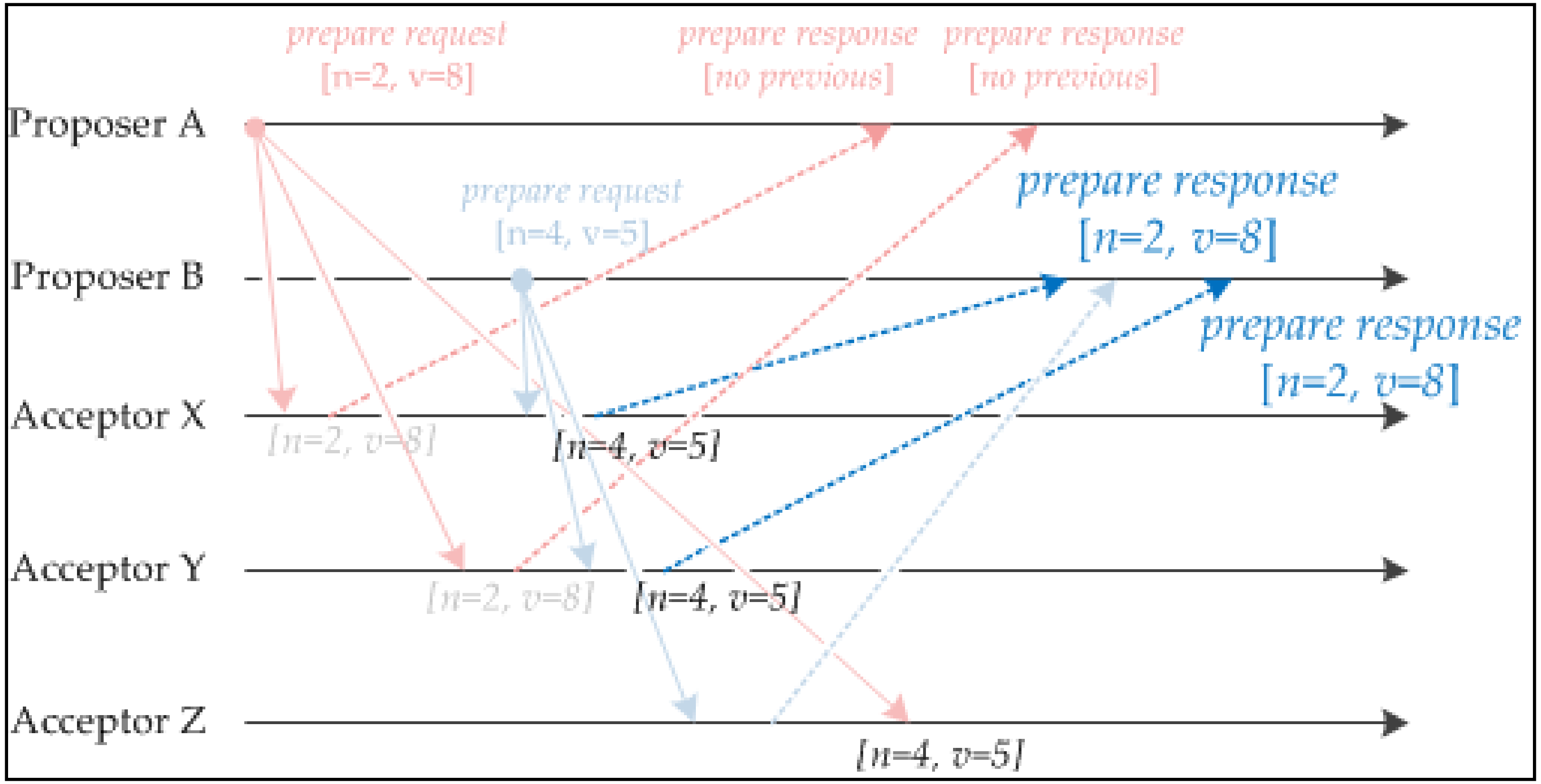
- Los aceptadores del quorum de la propuesta N de P pueden recibirla (aunque podría haber fallos)
- Dado un aceptador A que ha recibido la propuesta N, si el N es mayor que cualquier número de propuesta recibido anteriormente, entonces A debe devolver a P la promesa de ignorar todas las propuestas futuras que tengan un número menor que N
- Si A había aceptado una propuesta N' en algún momento del pasado, debe adjuntar a su promesa a P: el número N' y el valor v' correspondiente a la propuesta N'

Fase 1b: Prometer



https://medium.com/@angusmacdonald/paxos-by-example-66d934e18522

Fase 1b: Prometer

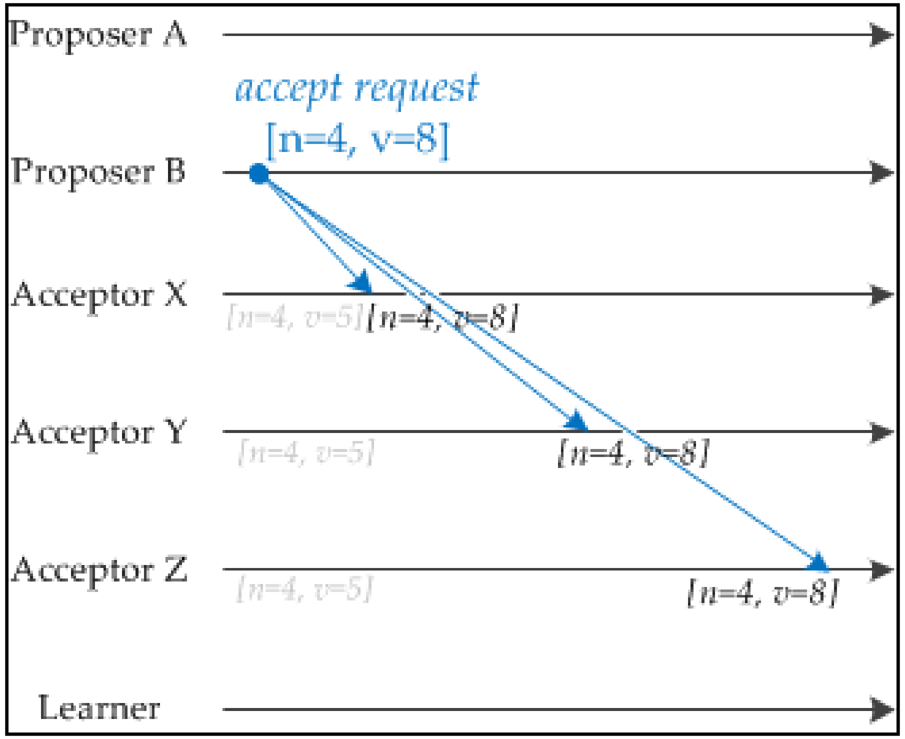


https://medium.com/@angusmacdonald/paxos-by-example-66d934e18522

Fase 2a: Aceptar petición

- Si P recibe suficientes promesas de su quórum de aceptantes (*una mayoría sobre el total de aceptadores*), debe decidir un valor y adjuntarlo a un mensaje de petición de aceptación
- Si alguno de los aceptadores había aceptado previamente alguna propuesta, entonces P está recibiendo sus valores en las promesas
- Si es el caso, P debe establecer el valor de su propuesta en el valor v' asociado con el número de propuesta N' más alto informado por los aceptadores
- Si ninguno de los aceptadores había aceptado una propuesta N', entonces P podrá escoger cualquier valor

Fase 2a: Aceptar petición

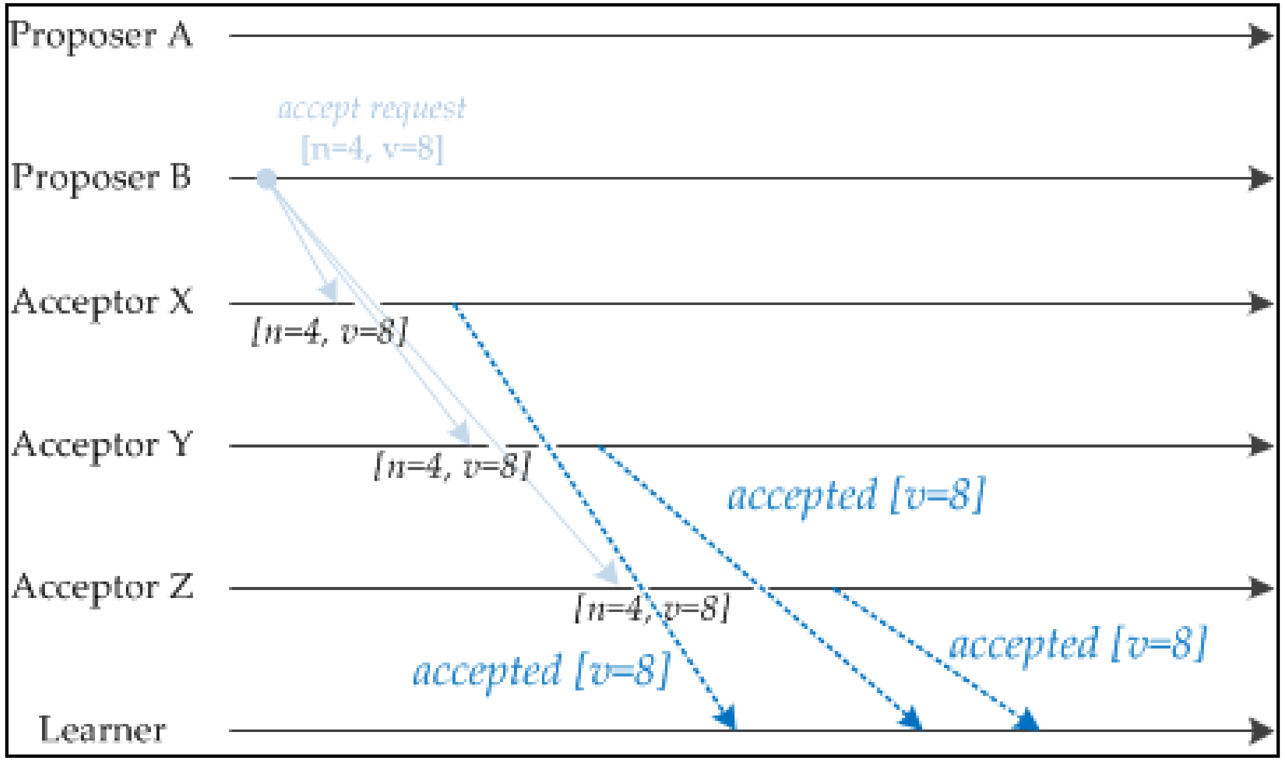


https://medium.com/@angusmacdonald/paxos-by-example-66d934e18522

Fase 2b: Aceptación

- Si un aceptador A recibe un mensaje de solicitud de aceptación para una propuesta N
    - Si ya se ha comprometido a considerar sólo las propuestas que tengan algún identificador N' mayor que N:
        - A ignora la solicitud de aceptación
    - Si no se ha comprometido a considerar ninguna propuesta relativa a un identificador N' superior a N:
        - A registra el valor correspondiente v asociado a la propuesta N
        - A envía un mensaje de aceptación al proponente P y a todos los aprendices

Fase 2b: Aceptación



https://medium.com/@angusmacdonald/paxos-by-example-66d934e18522

Propiedades de Paxos

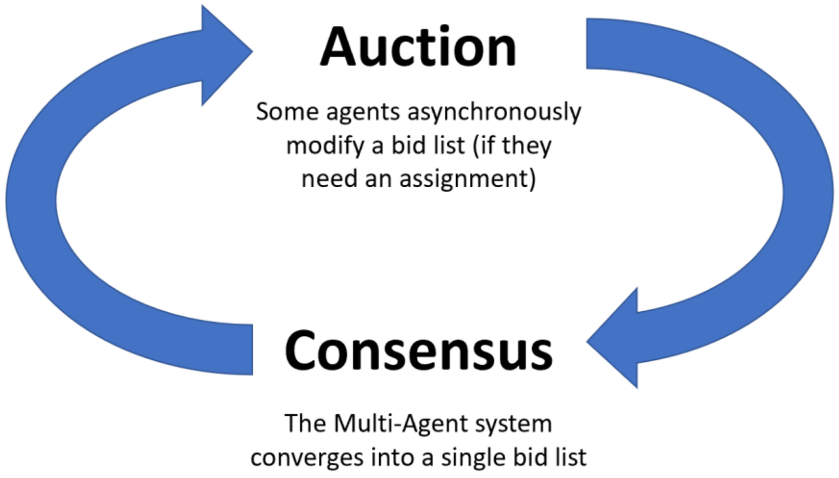
- Paxos permite
  - Agentes que operan a una velocidad arbitraria
  - Agentes que experimentan errores
  - Comunicación asíncrona
  - Fallos en la comunicación: los mensajes pueden perderse, desordenarse o duplicarse
- Los agentes con almacenamiento persistente pueden volver a unirse al sistema después de fallar
  - Paxos incluye un procedimiento de recuperación automática
- La consistencia en el valor consensuado está garantizada siempre y cuando
  - Si un número F de agentes fallan, ha de haber $2F + 1$ agentes funcionando correctamente
  - Los agentes pueden enviar mensajes a cualquier otro agente
  - Los mensajes se entregan sin daños
  - Los errores bizantinos no suceden

Usos prácticos de los algoritmos de consenso

- Paxos, Raft (https://raft.github.io) y otros se utilizan en muchos tipos de aplicaciones distribuidas, desde bases de datos hasta vehículos aéreos no tripulados
  - En general, son adecuados para escenarios distribuidos y asíncronos en los que se asume un error ocasional
- Al igual que los DCOPs o, hasta cierto punto, MA-PDDL o MA-A*, no permiten mucha autonomía de los agentes de forma predeterminada
- Sin embargo, no requieren de implementaciones ineficientes o muy complejas
- Los fracasos bizantinos se pueden resolver, por ejemplo, con ciertas extensiones de Paxos o con técnicas de encriptación (Blockchain)

Ejemplo: CBAA

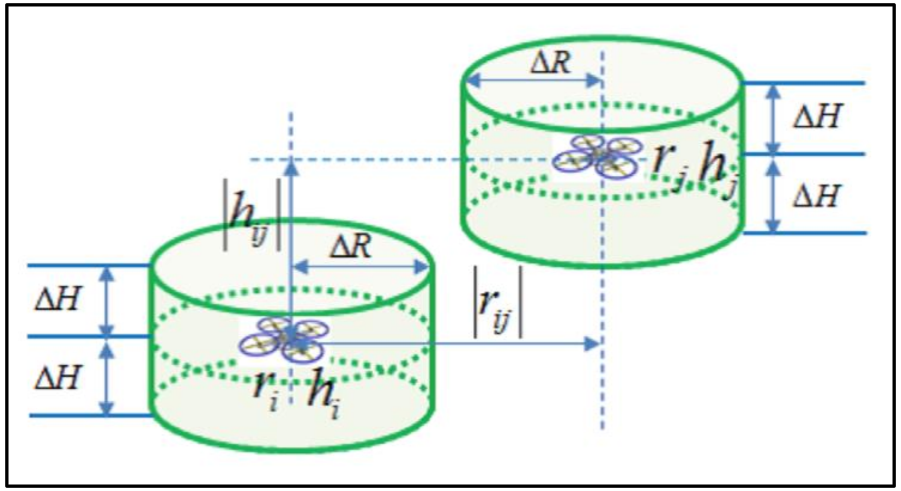
- CBAA: Consensus-based bundle algorithm
  - Choi, H. L., Brunet, L., & How, J. P. (2009). *Consensus-based decentralized auctions for robust task allocation*. IEEE transactions on robotics, 25(4), 912-926. [pdf]
- Objetivo: coordinar (de forma descentralizada) una flota de vehículos autónomos



- Combinación de dos algoritmos descentralizados:
  - Subasta
  - Consenso
- Garantías (bajo ciertas restricciones fácilmente asumibles):
  - Convergencia
  - Asignación libre de conflictos
  - Recompensa global ≥ (0,5 * óptimo teórico)

Ejemplo: prevención de colisiones

- Kuriki, Yasuhiro, and Toru Namerikawa. *Consensus-based cooperative formation control with collision avoidance for a multi-UAV system.* 2014 American Control Conference. IEEE, 2014.
- Para evitar colisiones, el líder (que calculará las rutas de todos) se elige por consenso



Ejemplo: selección de antena

- Muñoz, Filiberto et al. *Adaptive consensus algorithms for real-time operation of multi-agent systems affected by switching network events.* International Journal of Robust and Nonlinear Control 27, no. 9 (2017): 1566-1588.