Sistemas Multiagente

Tema 2.2: Negociación:Votaciones y Subastas.

Authors: Vicent Botti, Vicente Julián

Contexto

Coordinación

Cooperación

Negociación

+ objetivo global ⟷ + objetivo individual

¿Qué es la negociación?

¿Qué es la negociación?

- “Acción y efecto de negociar. Tratar y comerciar, comprando y vendiendo o cambiando géneros, mercancías, o valores para aumentar el caudal”. *Real Academia Española de la Lengua*

- “Un proceso por el cual una decisión conjunta es tomada por dos o más partes. Las partes primero verbalizan demandas contradictorias y se mueven hacia un acuerdo mediante un proceso de concesión, haciendo o buscando nuevas alternativas”. *Negotiation Behaviour*. D.G. Pruitt (1981)

- “La negociación es una forma de interacción en la que un grupo de agentes o personas con intereses en conflicto y un deseo de cooperar, intentan alcanzar un acuerdo mutuamente satisfactorio en la división de una serie de recursos limitados”. *Commitment in Dialogue: Basic concepts of interpersonal reasoning*. Walton y Krabbe (1995)

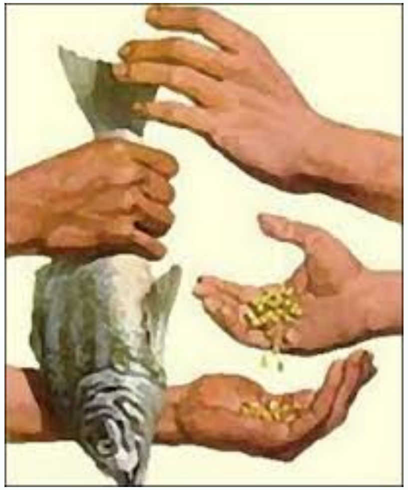
¿Qué es la negociación?

- Nuestro punto de vista:
  - Situación de conflicto, diferentes preferencias de los participantes → Sino, no hay necesidad de negociar
  - Interacción entre dos o más partes → Uno no necesita negociar consigo mismo
  - Es un proceso de búsqueda conjunta → Buscamos un acuerdo mutuamente satisfactorio para todos
  - Existe un inherente sentimiento de cooperación y competición → Naturaleza dual
  - Podemos encontrarla prácticamente en cualquier dominio

Un poco de historia: Hacia la negociación automática

- Negociación en la sociedad

Del truque al mercado. Aprendeconomia. Eva Baena. 2011



Un poco de historia: Hacia la negociación automática

- Negociación en la sociedad

Un poco de historia: Hacia la negociación automática

- Negociación en el mundo académico
  - Ingeniería informática
    - Los humanos son buenos reconociendo patrones pero malos para el cálculo → La mayoría de los humanos son malos negociadores
    - Las computadoras son buenas para el cálculo y están empezando a ser buenas reconociendo patrones
    - ¿Porqué no hacer que las computadoras negocien por nosotros? → **Negociación automática**

you are leaving money on the table

Un poco de historia: Hacia la negociación automática

- Negociación automática
  - Surge como rama de la inteligencia artificial
  - Es demasiado costoso buscar la solución óptima
    - Soluciones cercanas al óptimo a menor coste
  - Se suelen evitar las asunciones de:
    - Conocimiento Perfecto
    - Recursos computacionales ilimitados
    - Proceso no acotado en el tiempo
  - Los primeros trabajos datan de la época de los ’80
  - Área de investigación abierta

Aplicaciones

Comercio electrónico

- Cadenas de suministro
- Compras en grupo
- B2B
- Administración pública (ej: mercados de regantes)

Coordinación

- Tracking con cámaras de seguridad

Control

- Manejo de robot con procesos de negociación

Sistemas de apoyo a la decisión/negociación

- Equipos de negociación

Hay muchas formas de hacer **negociación**:

- Simple: Protocolo de Contrato (ya comentado)
- Subastas → como solución a la asignación de tareas o recursos
- Social Choice: Votaciones
- Negociación bilateral y multiparticipante
- Teoría de juegos: búsqueda de equilibrio

→ Negociación Automática y Resolución de Conflictos

Tomando decisiones en grupo: Social Choice

Introducción al social choice

Nociones básicas

Votaciones

¿Porqué tantos métodos de elección social?

Biblibografia

A

Opción B

Opción C

6.2.1Introducción al Social Choice

- ¿Qué es el social choice?

Introducción al Social Choice

- ¿Qué es el social choice?
  - Un grupo personas o agentes
    - Cada uno tiene sus propias preferencias
  - Objetivo: Tomar una decisión conjunta (tomar una decisión en grupo – group decision making)
    - Ejemplo clásico de la teoría de social choice son las votaciones
    - Ej: Elegir un presidente, elegir el mejor curso de acción, etc.
  - El social choice estudia la forma de agregar las preferencias de las personas.
    - Formalmente la cuestión es como combinar las preferencias para obtener resultados sociales.

Introducción al Social Choice

- Hombre y sociedad → Tomar decisiones en grupo es algo natural
- Los métodos estudiados en social choice surgen de forma natural en la sociedad
- Ha sido estudiado por las matemáticas y la teoría de juegos principalmente

Introducción al Social Choice

- Un poco de historia

Roma y Grecia
800AC-476DC

Ramón Llull
1232-1315DC

Marqués de Condorcet
1743-1794DC

Jean-Charles de Borda
1733-1799DC

Kenneth Arrow
1921DC

Introducción al Social Choice

- ¿Qué interés tiene el social choice desde el punto de vista computacional?

6.2.2 Nociones básicas

- Un conjunto de agentes $N = \{1,2,\ldots,n\}$ o votantes, estas son las entidades que expresarán sus preferencias
- Un conjunto de posibilidades o candidatos $O = \{o_1, o_2, \ldots, o_o\}$, si |O|= 2 tenemos una elección de parejas
- Relaciones de preferencia $\succeq_i$
  - Preferencia estricta $o_1 \succ_i o_2$ ← Asumimos esta
  - Preferencia débil $o_1 \succeq_i o_2$
  - Indiferencia $o_1 \sim_i o_2$
- La relación de preferencia induce un orden de preferencias sobre ***O*** para cada uno de los agentes.
- $\boldsymbol{F_N}$ es la función de social choice o criterio del ganador

Nociones básicas

- Preferencias:
  - Cada votante tienen sus preferencias sobre las distintas opciones (O) ⇒ una ordenación sobre el conjunto de posibles opciones.
  - Ejemplo: sea el siguiente conjunto de opciones

    O = {ginebra, ron, brandy, whisky]

    Podríamos tener el agente ccc con el siguiente orden de preferencias

    $$o_{ccc} = (\text{whisky, brandy, ron, ginebra})$$

    que expresaríamos:

    $$\text{whisky} \succ_{ccc} \text{brandy} \succ_{ccc} \text{ron} \succ_{ccc} \text{ginebra}$$

Nociones básicas

- Agregación de preferencias

  - El problema fundamental de la teoría de social choice es:

    dada una colección de órdenes de preferencia, una para cada votante, ¿cómo podemos combinarlas para obtener una decisión de grupo, que refleje lo más fielmente posible las preferencias de los votantes?

  - Tenemos dos variantes de agregación de preferencias:
    - Funciones de bienestar social (social welfare functions)
    - Funciones de elección social (social choice functions)

Nociones básicas

- Funciones de bienestar social (social welfare functions)
  - Sea ∏(O) el conjunto de ordenaciones de preferencia sobre O
  - Una función de bienestar social determina, a partir de las preferencias de los votantes, un orden de preferencia social:

$$F_N : \underbrace{\Pi(O) \times \ldots\ldots\ldots \times \Pi(O)}_{n\ veces} \rightarrow \Pi(O)$$

  - Denotado por el resultado de una función de bienestar social
  - Ejemplo: concursos de belleza

Nociones básicas

- Funciones de selección social (social choice functions)

  - Algunas veces solo necesitamos seleccionar una de las posibles opciones en vez de establecer un orden social
  - Una función de selección social determina, a partir de las preferencias de los votantes, una de las opciones:

$$F_N : \underbrace{\Pi(O) \times \ldots\ldots\ldots \times \Pi(O)}_{n\ veces} \rightarrow O$$

  - Ejemplo: elección presidencial

Nociones básicas

- Condiciones Condorcet
  - Una opción ***o*** es ganadora Condorcet si para cualquier otra opción ***o'***, el número de agentes que ***o*** $\succcurlyeq$ ***o'*** es mayor o igual que el número de agentes que ***o'*** $\succcurlyeq$ ***o***
  - Una opción ***o*** es perdedora Condorcet si para cualquier otra opción ***o'***, el número de agentes que ***o'*** $\succcurlyeq$ ***o*** es mayor o igual que el número de agentes que ***o*** $\succcurlyeq$ ***o'***
  - $F_N(.)$ es condorcet ganador si escoge como ganador al ganador condorcet
  - $F_N(.)$ cumple el criterio condorcet perdedor si excluye al condorcet perdedor como ganador

Nociones básicas

- Supongamos 3 agentes Ag= {Carlos, Juan y Luisa} y tres opciones O={a, b, c}, y las siguientes preferencias (Condiciones Condorcet)

$$\begin{array}{llllll} Carlos & a \succ & b \succ & c & \equiv & a \succ_{Carlos} b \succ_{Carlos} c \\ Juan & b \succ & c \succ & a & \equiv & b \succ_{Juan} c \succ_{Juan} a \\ Luisa & a \succ & c \succ & b & \equiv & a \succ_{Luisa} c \succ_{Luisa} b \end{array}$$

  - Ganador Condorcet: a
  - Perdedor Condorcet: c

Nociones básicas

- Paradoja Condorcet

$$\begin{array}{llll} Carlos & a \succ & b \succ & c \\ Juan & b \succ & c \succ & a \\ Luisa & c \succ & a \succ & b \end{array}$$

- Ganador Condorcet: No hay!!
  - Para cada posible opción (candidato) hay otra opción que es preferida por una mayoría de votantes
- ¿A quien hacemos ganador?
- Esto es la paradoja Condorcet: hay situaciones en las que, sin importar el resultado que elijamos, una mayoría de los votantes no estarán contentos con el resultado elegido.

Nociones básicas

- Propiedades deseables en procedimientos de votación
  - ¿Podemos clasificar las propiedades que deseamos que satisfaga un buen procedimiento de votación?
  - Destacaremos dos propiedades clave:
    - La propiedad de Pareto
    - Independencia de alternativas irrelevantes (IIA)

Nociones básicas

https://www.youtube.com/watch?v=8HEXASmUvzM

https://www.youtube.com/watch?v=20UruaP-VC0

- Criterio Pareto
  - Si para todo votante ***i***, ***o*** $\succ_i$ ***o'***, entonces $F_N(O) \neq o'$
  - ***Si todos prefieren a o más que a o', entonces o debe ser clasificado antes que o' en el resultado social.***
- Consistencia
  - Supongamos el espacio de votantes particionado en dos subgrupos $\boldsymbol{N=\{N_1,N_2\}}$
  - Denotamos a la elección social de $\boldsymbol{N_1}$ como $\boldsymbol{F_{N1}(O)}$ y a la elección social de $\boldsymbol{N_2}$ como $\boldsymbol{F_{N2}(O)}$
  - Si $F_{N_1}(O) \bigcap F_{N_2}(O) \neq \emptyset \rightarrow F_N(O) = F_{N_1}(O) \bigcap F_{N_2}(O)$

Nociones básicas

- Independencia de alternativas irrelevantes (IIA)

  Si o_i se clasifica por encima de o_j en el resultado social debe depender solamente de los ordenamientos relativos de o_i y o_j en los perfiles de los votantes.

Nociones básicas

- Monotonía
  - Suponer que ***P*** es el perfil de preferencias de ***N*** y $F_N(O|P)=0$
  - ***P'*** es un perfil de preferencias basado en ***P*** donde la posición de ***o*** ha sido mejorada en al menos una posición
  - $F_N(.)$ es monótono si $F_N(O|P')=o$

Votaciones

- La votación es el mecanismo de social choice por excelencia
- Distintos tipos:
  - Comparaciones por pares:
    - Método de Dodgson
    - Regla de Copeland
  - Posicionales:
    - Pluralidad
    - Conteo Borda
  - Posicionales + Comparación por pares
    - Pluralidad a dos rondas
    - Sistema de Hare
  - Basados en agenda:
    - Proceso de enmienda
    - Proceso sucesivo

Votaciones

Comparación por pares:

- Método de Dodgson (Lewis Carroll)
  - Elige el ganador condorcet, y en caso que no exista busca el candidato más próximo al ganador condorcet → Menor número de cambios (intercambios entre candidatos contiguos) en las preferencias para convertirlo en ganador condorcet
  - Requiere el recorrido completo del ránking de los agentes
  - Problema NP-duro

Votaciones

- Comparación por pares:
  - Método de Dodgson

$$Agentes:5 \quad a \succ \quad b \succ \quad c$$
$$Agentes:4 \quad b \succ \quad c \succ \quad a$$
$$Agentes:3 \quad c \succ \quad a \succ \quad b$$

- No hay ganador condorcet
  - Ganador de Dodgson→ Hay que aplicar dos intercambios cambios a ***a (en dos agentes intercambiar las preferencias entre a y c)***

Votaciones

Comparación por pares:

- Regla de Copeland
  - Busca la alternativa ***o*** cuyo índice copeland ***IC(o)*** es mayor

$$\arg\max_{o \in O} IC(o) = \arg\max_{o \in O} \sum_{\substack{o' \neq o \\ o' \in O}} p(o, o')$$

$$p(o,o') = \left\{ \begin{array}{cc} 1 & \#(o \succ o') > (o' \succ o) \\ 0 & otherwise \end{array} \right\}$$

  - Recorrido completo de las preferencias

Votaciones

- Comparación por pares:
  - Regla de Copeland

$$
\begin{array}{llll}
Agentes: 5 & a \succ & b \succ & c \\
Agentes: 4 & b \succ & c \succ & a \\
Agentes: 3 & a \succ & c \succ & b
\end{array}
$$

  - ***IC(a)=1+1=2***
  - ***IC(b)=1***
  - ***IC(c)=0***

Votaciones

- Posicionales: pluralidad
  - Función de elección social: selecciona una única opción
  - Cada votante presenta sus preferencias
  - Cada candidato recibe un punto por cada orden de preferencia en que ocupa el primer lugar.
  - El ganador es el que tiene mayor número de puntos.
  - Es necesaria una regla para romper desempates!
  - Solo requiere enviar la mejor opción
  - Ejemplo: Las elecciones políticas en el Reino Unido.
  - Si tenemos sólo dos candidatos, entonces la pluralidad es una simple elección de mayoría simple.

Votaciones

- Posicionales: Pluralidad

$$Agentes:5 \quad a \succ \quad b \succ \quad c$$
$$Agentes:4 \quad b \succ \quad c \succ \quad a$$
$$Agentes:3 \quad a \succ \quad c \succ \quad b$$

  - ***Votos a: 8***
  - ***Votos b: 4***
  - ***Votos c: 0***

Votaciones

- Anomalías con la pluralidad:
  - Suponer que tenemos 100 agentes ($|Ag|=100$) y tres opciones $O=\{o_1, o_2, o_3\}$ con:

    40% de votantes que votan por o_1

    30% de votantes que votan por o_2

    30% de votantes que votan por o_3

  - Con la pluralidad o_1 es la opción elegida a pesar de que una amplia mayoría (60%) prefieren a otra opción/candidato!

Votaciones

- Posicionales:
  - Una razón por la que la pluralidad tiene tantas anomalías es que ignora la mayor parte de las órdenes de preferencias del votante: sólo considera al candidato mejor clasificado.
  - El conteo de Borda considera un orden de preferencia total.
  - Conteo Borda
    - Cada agente asigna una puntuación a cada candidato igual a su posición en su ránking de preferencias (k) descendente menos 1.

      Para cada candidato/opción se va contando la fuerza de la opinión a favor de dicho candidato. Si o_i aparece primero en un orden de preferencia, entonces se incrementa el conteo de o_i en k -1, a continuación se incrementa el conteo para el siguiente candidato en ese orden de preferencia en k - 2,. . . Hasta que el candidato final en el orden de preferencia incrementa su conteo en 0.
    - Después de hacer esto para todos los votantes, el candidato con mayor número de puntos es seleccionado como ganador
    - Es un criterio de semi-unanimidad
    - Requiere revelar todas las preferencias

Votaciones

- Posicionales:
  - Conteo Borda

$$
\begin{array}{llll}
Agentes:5 & a\succ & b\succ & c \\
Agentes:4 & b\succ & c\succ & a \\
Agentes:3 & a\succ & c\succ & b
\end{array}
$$

- ***Votos a: 5*2 + 4*0+3*2=10+6=16***
- ***Votos b: 5*1+4*2+0*3=5+8=13***
- ***Votos c: 5*0+4*1+3*1=7***

Votaciones

- Posicionales:
  - Paradoja de Borda

$$\begin{array}{llll} Agentes:1 & a \succ & b \succ & c \\ Agentes:7 & a \succ & c \succ & b \\ Agentes:7 & b \succ & c \succ & a \\ Agentes:6 & c \succ & b \succ & a \end{array}$$

  - Ganador pluralidad: a
    - Problema:
      - a>b → 8   b>a→13
      - a>c → 8   c>a→13
      - b>c → 8   c>b→13
  - Ganador borda
    - Votos a: 1*2+7*2=2+14=16
    - Votos b: 1*1+7*2+6*1=1+14+6=21
    - Votos c: 7*1+7*1+6*2=7+7+12=14+12=26

Votaciones

- Posicionales+Comparación por pares:
  - Pluralidad a dos rondas
    - Cada agente vota a su mejor candidato.
    - Los dos candidatos con mayor número de votos permanecen para una próxima votación. El resto son eliminados.
    - Se vuelve a votar con solo los dos candidatos ganadores como opciones

Votaciones

- Posicionales+Comparación por pares:
  - Pluralidad a dos rondas

$$\begin{array}{llll} Agentes:5 & a \succ & b \succ & c \\ Agentes:4 & b \succ & c \succ & a \\ Agentes:3 & c \succ & b \succ & a \end{array}$$

- ***Votos a: 5 → Votos a: 5***
- ***Votos b: 4 → Votos b: 4+3=7***
- ***Votos c: 3***

Votaciones

- Posicionales+Comparación por pares:
  - Sistema de Hare
    - Cada agente vota a su mejor candidato.
    - Si uno de los candidatos recibió la mayoría de los votos, es ganador
    - Sino, se elimina el candidato con menor número de votos y se repite el proceso hasta que uno de los candidatos gana por mayoría

Votaciones

- Posicionales:+Comparación por pares
  - Sistema de Hare

$$Agentes:1 \quad a \succ \quad b \succ \quad c$$
$$Agentes:7 \quad a \succ \quad c \succ \quad b$$
$$Agentes:7 \quad b \succ \quad c \succ \quad a$$
$$Agentes:6 \quad c \succ \quad b \succ \quad a$$

  - Ganador Hare
    - Votos a: 8 → Votos a: 8
    - Votos b: 7 → Votos b: 7+6=13
    - Votos c: 6

Votaciones

- Basados en agenda: elecciones por pares secuenciales lineales
  - Aquí elegimos una ordenación de los resultados – la agenda- que determina quién juega contra quién.

    Alguien escoge un orden para los candidatos

    Ej: a→b→c→d

    la primera elección es entre *a* y *b*, y el ganador pasa a una elección con *c*, y el ganador de esta elección entra en una elección con *d*
  - Proceso de enmienda
    - Los candidatos son enfrentados a pares hasta que una única opción queda como ganadora
  - Proceso sucesivo
    - En orden, cada uno de los candidatos es enfrentado contra el resto. Si obtiene la mayoría es ganador. Sino, es eliminado del conjunto

Votaciones

- Basados en agenda

$$\begin{array}{lllll}
Agentes:3 & a\succ & b\succ & c\succ & d \\
Agentes:7 & a\succ & c\succ & b\succ & d \\
Agentes:7 & d\succ & a\succ & c\succ & b \\
Agentes:6 & c\succ & b\succ & d\succ & a
\end{array}$$

- Agenda: d→a→b→c
  - Sucesivo: d vs (a,b,c) →a vs (b,c)
  - Enmienda: d vs a →d vs b → b vs c

Votaciones

- Basados en agenda: anomalías

$$
\begin{array}{lllll}
Agentes: 3 & a \succ & b \succ & c \succ & d \\
Agentes: 7 & a \succ & c \succ & b \succ & d \\
Agentes: 7 & d \succ & a \succ & c \succ & b \\
Agentes: 6 & c \succ & b \succ & d \succ & a
\end{array}
$$

- Agenda: c→b→a→d
  - Sucesivo: c vs (a,b,d)→ b vs (a,d)→a vs d
  - Enmienda: c vs b → c vs a → a vs d
- El ganador depende del orden de la agenda!!!

Votaciones

- Basados en agenda: anomalías
  - Suponer:
    - 33 votantes tienen preferencias

      $o_1 \succ_i o_2 \succ_i o_3$
    - 33 votantes tienen preferencias

      $o_3 \succ_i o_1 \succ_i o_2$
    - 33 votantes tienen preferencias

      $o_2 \succ_i o_3 \succ_i o_1$
  - Luego, para cada candidato, ¡se puede establecer una agenda para que gane dicho candidato en una elección por pares secuencial!

Votaciones

- Grafos de mayorías:

  - Los grafos de mayorías nos permiten ilustrar fácilmente como establecer una agenda para que gane un candidato.

  - ¿Qué es un grafo de mayorías?
    - Un grafo dirigido con:
      - Vertíces = candidatos
      - Un arco (i,j) si *i* ganaría a *j* es una elección por mayoría simple
    - Una representación compacta de las preferencias de los votantes.

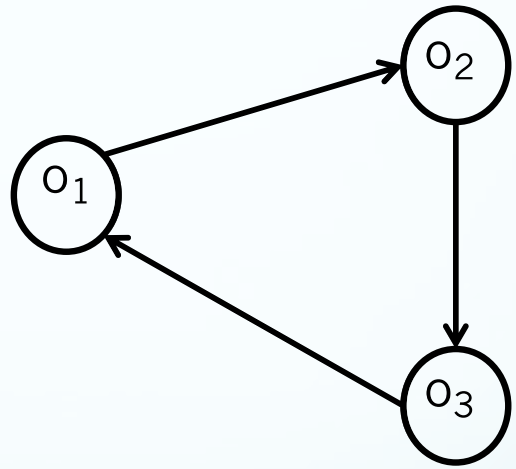
Votaciones

- Grafos de mayorías: ejemplo
  - 33 votantes tienen preferencias

    $o_1 \succ_i o_2 \succ_i o_3$
  - 33 votantes tienen preferencias

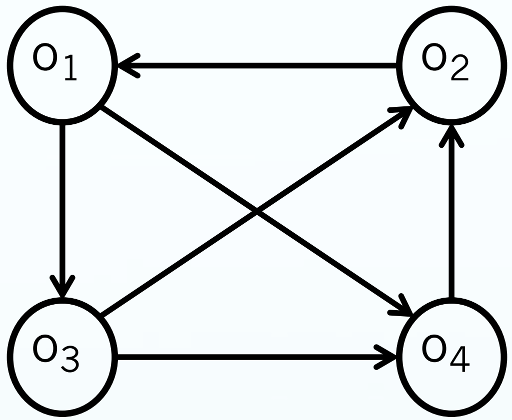
    $o_3 \succ_i o_1 \succ_i o_2$
  - 33 votantes tienen preferencias

    $o_2 \succ_i o_3 \succ_i o_1$
    - con la agenda (o_3, o_2, o_1) vence o_1
    - con la agenda (o_1, o_3, o_2) vence o_2
    - con la agenda (o_1, o_2, o_3) vence o_3



Votaciones

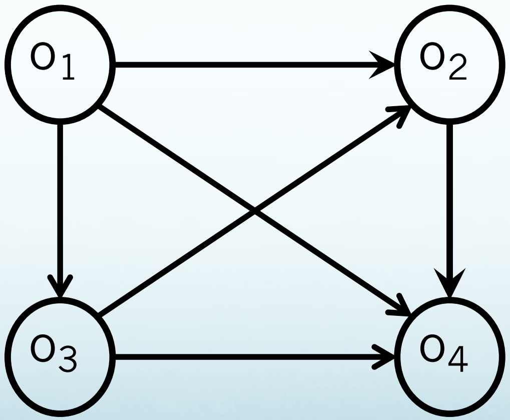
- Otro ejemplo de grafo de mayorías



- Para cada uno de los candidatos del grafo dar una agenda para que venza.

Votaciones

- Vencedores Condorcet
  - Un ganador de Condorcet es un candidato que ganaría a todos los candidatos en una elección por parejas.
  - En este grafo o_1 es un ganador Condorcet.



6.2.4 ¿Por qué tantos sistemas de elección social?

Sistema	a	b	c	d	e
Enmienda	1	1	1	0	0
Sucesivo	0	1	1	0	0
Copeland	1	1	1	1	0
Dodgson	1	0	0	1	0
Pluralidad	0	0	1	1	1
Borda	0	1	1	1	1
Pluralidad 2 ron.	0	1	0	1	0
Hare	0	1	0	1	0

Abreviatura	Criterio
a	Ganador Condorcet
b	Perdedor Condorcet
c	Monotonía
d	Criterio Pareto
e	Consistencia

¿Por qué tantos sistemas de elección social?

- ¡No pasa nada, voy a diseñar el proceso de votación justo y universal!

- Lo siento, pero tengo una mala noticia que darte…

¿Por qué tantos sistemas de elección social?

- Teorema de Arrow
  - Es el resultado más importante de social choice de los últimos tiempos
  - Función de bienestar social ***W***→ Da un ránking social de las alternativas
  - Criterios de justos y lógicos para Arrow
    - **Criterio Pareto (P(W))** : Si todos los agentes están de acuerdo en el orden entre dos candidatos, W debe seleccionar ese orden
    - **Independencia de alternativas irrelevantes (IIA(W))**: El orden entre dos candidatos debería depender solo del orden relativo de los candidatos dado por los agentes
    - **No dictatorial (¬D(W))**: No existe un solo agente cuyas preferencias siempre determinen el orden social.

¿Por qué tantos sistemas de elección social?

- Teorema de Arrow:

  Si |O|>3, P(W) , IIA(W)→D(W)

  - Para las elecciones con más de 2 candidatos, el único procedimiento de votación que satisface la condición de Pareto y IIA es una dictadura, en la que el resultado social es, de hecho, simplemente seleccionado por uno de los votantes.
  - Se trata de un resultado negativo: ¡hay límites fundamentales a la toma de decisiones democrática!
  - ¿Qué implicaciones tiene este teorema?
    - No podemos encontrar un orden social que cumpla las tres propiedades a la vez
    - En muchas ocasiones no existe forma justa y lógica de agregar las preferencias individuales → No existe forma exacta de determinar la voluntad colectiva
  - Pero ¿y si yo solo quiero seleccionar un ganador? ¿Puedo tener criterios de justicia y lógicos?

¿Por qué tantos sistemas de elección social?

- Lo siento, pero tengo otra mala noticia que darte…

¿Por qué tantos sistemas de elección social?

- Teorema de Muller-Satterthwaite
  - Función de elección social ***F***→ Elegir un ganador
  - Criterios justos y lógicos
    - **Criterio Pareto (P'(F))**
    - **Monotonicidad (M(F))**
    - **No dictatorial (¬D'(W))**: F es no dictatorial si no existe un agente i tal que la elección de F siempre seleccione el mejor candidato para i
  - Si |O|>3, P'(F) , M(F)→D'(F)

¿Por qué tantos sistemas de elección social?

- Manipulación estratégica mediante votación táctica:
  - En ocasiones, los votantes se pueden beneficiar al tergiversar estratégicamente sus preferencias, es decir, la mentira - voto táctico.
  - Suponer que tenemos las siguientes preferencias:

$$o_1 \succsim_i o_2 \succsim_i o_3$$

  - mientras que creemos que el 49% de los votantes tienen las siguientes preferencias

$$o_2 \succsim_i o_1 \succsim_i o_3$$

  - y creemos que el 49% tiene las preferencias

$$o_3 \succsim_i o_2 \succsim_i o_1$$

  - Haríamos mejor votando por o_2, a pesar de que este no es nuestro verdadero perfil de preferencias.
  - Esta es la votación táctica: un ejemplo de manipulación estratégica de los votos.

¿Por qué tantos sistemas de elección social?

- ¿Existen métodos de votación que no son manipulables, en el sentido de que los votantes no pueden beneficiarse de tergiversar las preferencias?
- La respuesta viene dada por el teorema de Gibbard-Satterthwaite:

  El único método de votación no manipulable que satisface la propiedad de Pareto para las elecciones con más de 2 candidatos es una dictadura.
- En otras palabras, todo método de votación "realista" es presa de la manipulación estratégica. . .

  ¡Complejidad computacional al rescate!
- Gibbard-Satterthwaite sólo nos dice que la manipulación es posible en principio. No da ninguna indicación de cómo tergiversar preferencias.
- Bartholdi, Tovey, y Trick demostraron que hay elecciones que son propensas a la manipulación, en principio, pero que la manipulación era computacionalmente compleja.
- ¡'El Voto Único Transferible" es NP-duro de manipular!

¿Por qué tantos sistemas de elección social?

- Todo lo que os he contado es social choice asumiendo veracidad de los agentes
- Cuando los agentes juegan estratégicamente, la cosa se complica todavía más → Podéis consultar material en la bibliografía
- Los sistemas de ránking (e.g., pagerank) son casos particulares de problemas de social choice → Más material en la bibliografía

BIBLIOGRAFÍA

- General:
  - *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Y. Shoham, K. Leyton-Brown. Ed. Cambridge (2010)
  - *Voting Systems for Social Choice*. H. Nurmi. Handbook of Group Decision and Negotiation (2010)
  - *A Short Introduction to Computational Social Choice*. Y. Chevaleyre, U. Endriss, J. Lang, N. Maudet. SOFSEM 2007: Theory and Practice of Computer Science (2007)
- Votación estratégica:
  - *The computational difficulty of manipulating an election.* J.J. Bartholdi III, C. A. Tovey, M. Trick. Social Choice and Welfare (1989)
  - Manipulation of voting schemes: a general result. A. Gibbard. Journal of the Econometric Society (1973)
- Sistemas de Ranking:
  - *Axiomatic Foundations for Ranking Systems.* A. Altman, M. Tennenholtz. Journal of Artificial Intelligence Research (2008)

¿Qué es una subasta?

An auction is a protocol that allows agents to indicate their interests in one or more resources and that uses these indications of interest to determine both an allocation of resources and a set of payments by the agents.

[Shoham & Leyton-Brown 2009]

¿Dónde se utilizan subastas?

- **Asignación de recursos (o tareas)**
  - Subastas del tesoro
  - Derecho a perforaciones de petróleo,
  - Uso del espectro radioeléctrico
  - Adquisición de bienes y servicios privados y/o públicos
  - Subastas en Internet
- Market-based computing

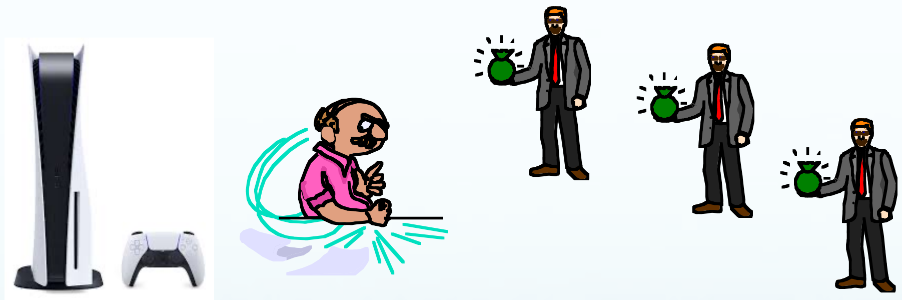
  Uso de un método "basado en el mercado", como una subasta, para calcular el resultado de un problema distribuido.
  - Control de aire acondicionado
  - Control de producción
  - Navegación en robots móviles
  - Redes de sensores
  - Transporte inteligente
  - Publicidad digital (Real-Time bidding)

¿Por qué hacer subastas?

- Ajuste de precios basado en el mercado → Para objetos de valor desconocido, el valor es evaluado dinámicamente por el mercado!
- Permiten asignar bienes, tareas, recursos ...
- Flexibles → Cualquier tipo de objeto
- Dinámicas→ Interactivo
- **Automatizables**
    - el uso de reglas simples reduce la complejidad de las negociaciones
    - ideal para su implementación
- *La maximización de los ingresos y las asignaciones de recursos/tareas eficientes son **objetivos alcanzables***

Más info sobre Subastas

- Participantes: subastador, postores
- **Acuerdo** forzoso entre el subastador y el/los licitador/es ganador/es
- Fácilmente implementable, por ejemplo a través de Internet
  - Muchos sitios de subastas en Internet
- Subasta (venta de item(s)): Un vendedor, múltiples compradores
  - Ej: vender una foto de una Playstation 5 en eBay

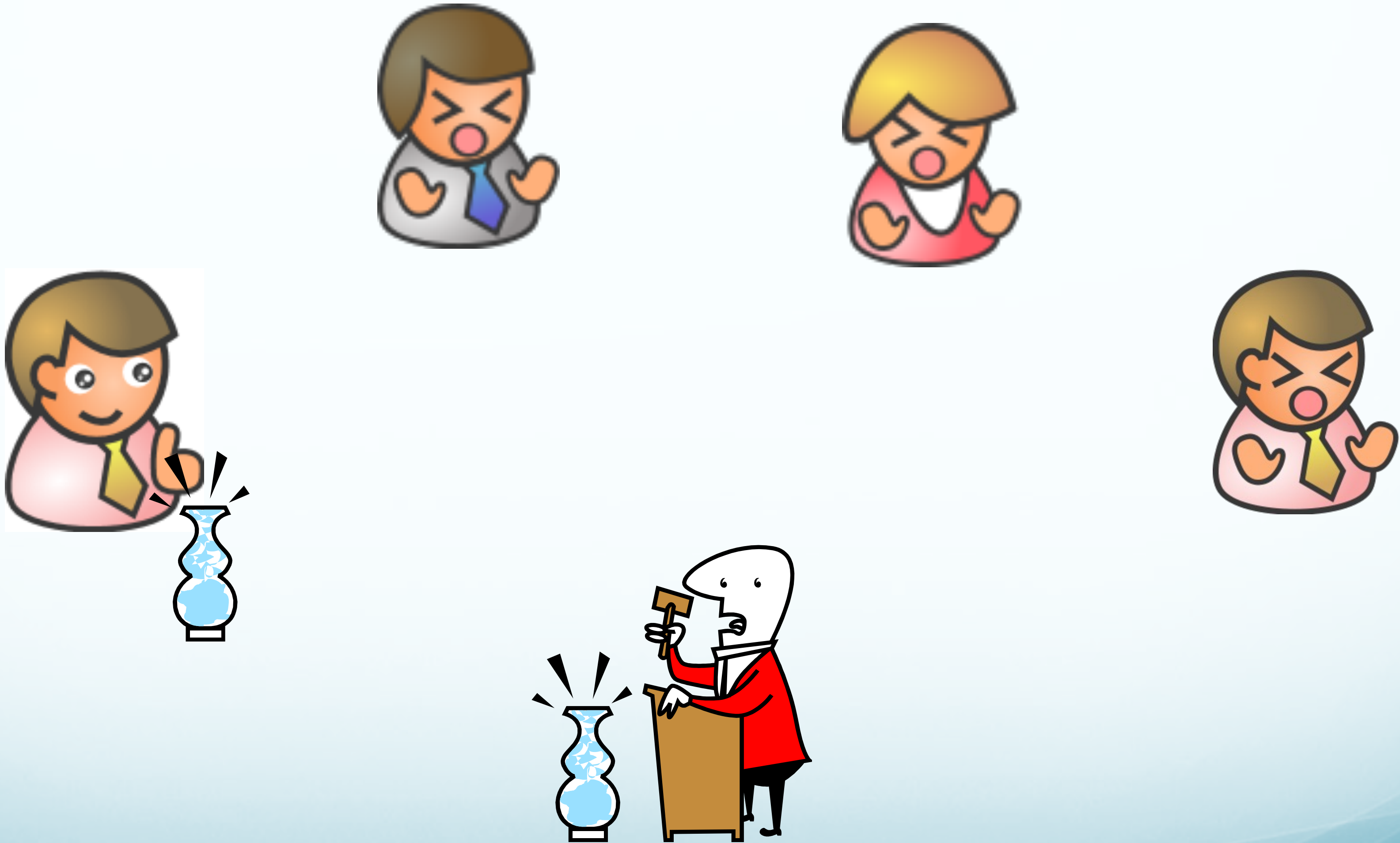


- Subasta inversa (comprando item(s)): Un comprador, múltiples vendedores
  - Ej: compras electrónicas (e-procurement)
- Discutiremos la teoría en el contexto de las subastas, pero la misma teoría se aplica a las subastas inversas

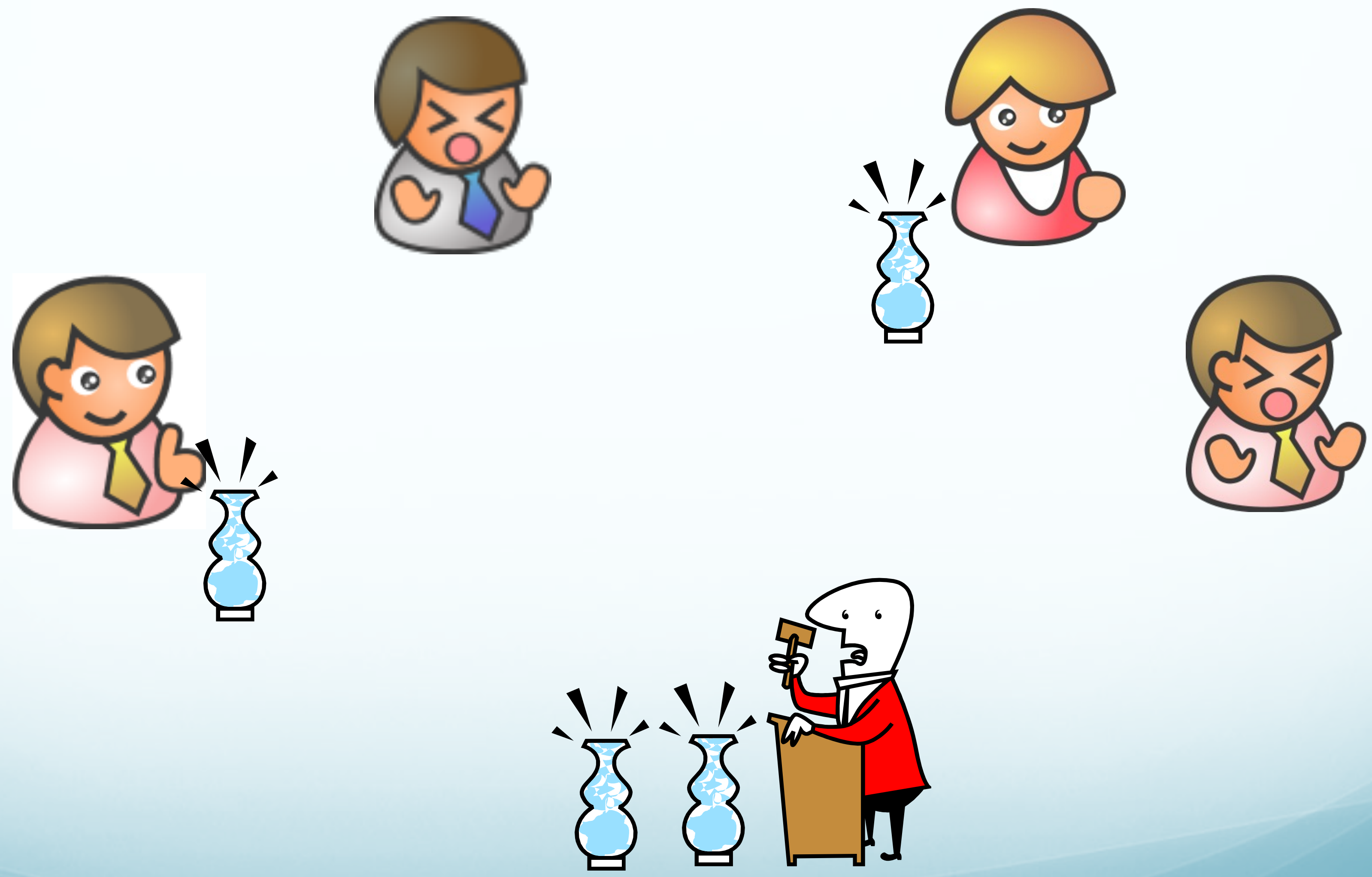
Subastas: Tipologías

- En general
  - Distribución de uno o de varios bienes discretos
  - Pueden ser del mismo tipo o de distinto tipo
  - Pueden ser Monoatributo vs Multiatributo
  - Pueden ser mediadas o no

Subastas de un solo bien



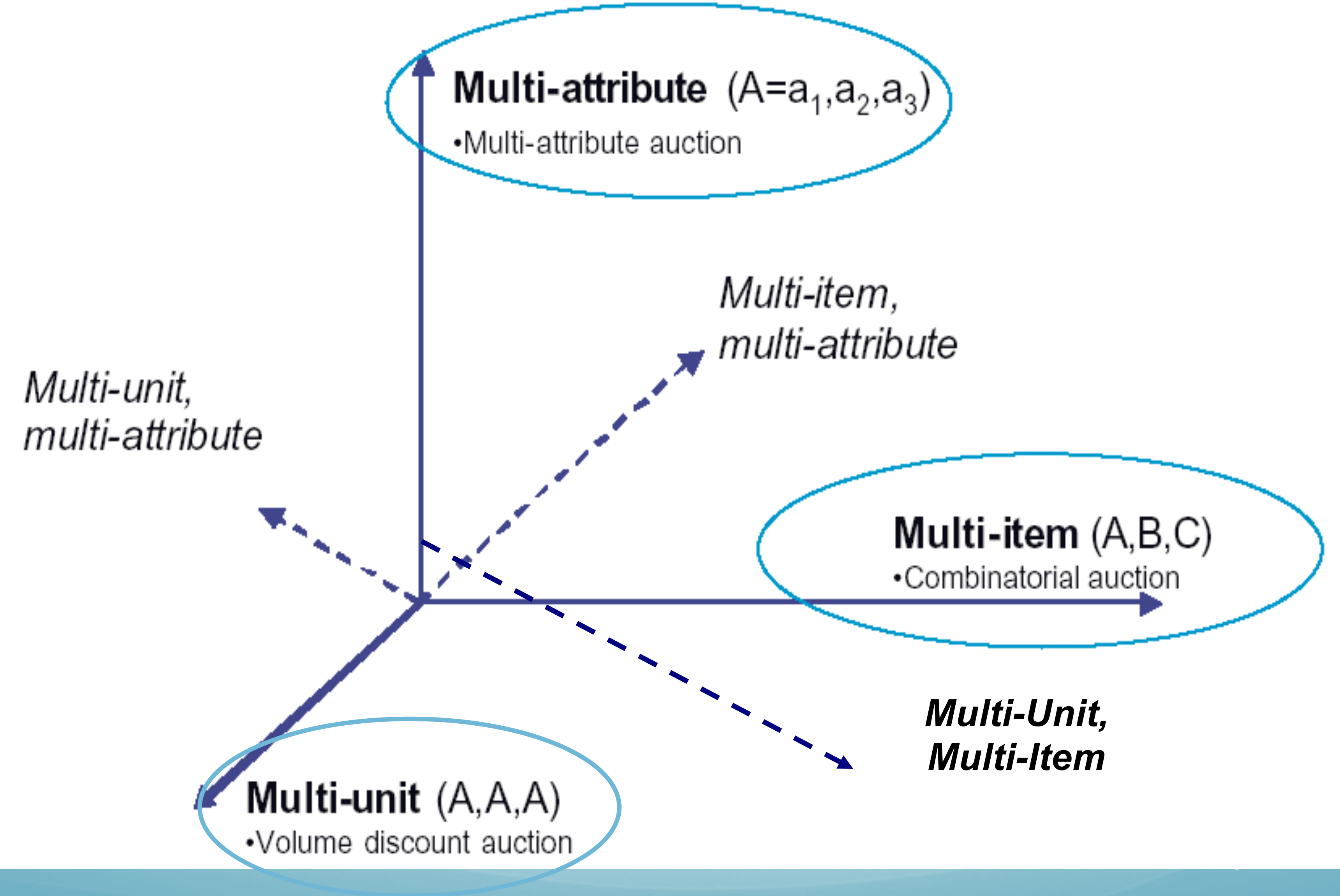
Subastas de multi-unidad



Subastas multi-artículos

Subastas multi-artículos (II)

Subastas: Dimensiones



Subastas por un solo bien

- Un vendedor y n compradores
- Negociación por un único bien o producto
- Cada comprador valora en mayor o menor grado el producto → Está dispuesto a pagar más o menos
- Los compradores prefieren pagar cuanto menos mejor
- Los vendedores prefieren cobrar cuanto más mejor
- ¿Qué estrategia seguir?: una estrategia es dominante cuando esta es la más provechosa independientemente de la estrategia de los otros jugadores

Subastas más comunes (de un solo item)

- Subasta inglesa
- Subasta japonesa
- Subasta holandesa
- Sobre cerrado de primer precio
- Sobre cerrado de segundo precio (Vickrey)

Subasta inglesa

- Tal vez el tipo más conocido de subasta (Sotheby's):
- Procedimiento:
  1. Subastador sugiere precio de reserva (puede ser cero)
  2. Los agentes deben ofertar más que la oferta más alta actual
  3. Todos los agentes visualizan las ofertas y pueden realizar ofertas en cualquier momento
  4. (No más ofertas) gana la oferta más actual alta y el agente ganador tiene que pagar la cantidad ofertada

  - Estrategia dominante: Pujar la unidad incremental mínima hasta que el resto de agentes alcancen su máximo
  - La maldición del ganador: ¿Por qué ningún otro agente valoraba el bien como yo? ¿Pagué demasiado?

Subasta japonesa

- De precio ascendente, como la inglesa
- Procedimiento:
  1. El vendedor indica un precio inicial
  2. A cada paso, todos los compradores indican de forma pública si están dispuestos a aceptar el precio. Los que no, abandonan
  3. El vendedor incrementa el precio hasta que solo queda 1 comprador
  4. El ganador paga el precio de la anterior ronda
- Estrategia dominante: Continuar en el proceso siempre que nuestro límite máximo lo permita. Decir la verdad es una estrategia dominante cuando nuestro precio máximo no depende de lo que hagan otros.

Subasta holandesa

- Procedimiento:
  1. El subastador comienza con un valor inicial artificialmente alto muy por encima del valor esperado.
  2. El subastador reduce continuamente el precio de oferta en un pequeño valor hasta. . .
  3. Algún agente hace una oferta por el bien igual al precio actual de oferta
  4. El agente ganador tiene que pagar la cantidad ofertada
- Este tipo de subasta también es susceptible a la maldición del ganador
- No tiene estrategia dominante

Sobre cerrado de primer precio

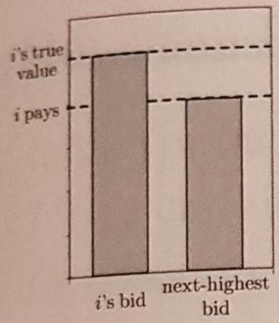
- La subasta de sobre cerrado de primer precio es la más simple de todas las subastas consideradas

- Procedimiento:
  1. Ronda única, en la que los ofertantes envían sus ofertas privadamente al subastador
  2. El subastador selecciona al agente con la oferta más alta
  3. El agente tiene que pagar la cantidad de su oferta

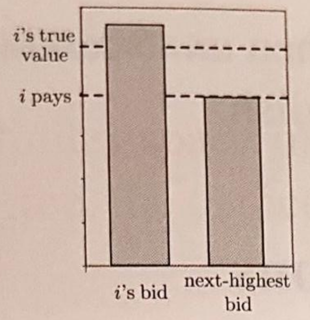
- Estrategia dominante: no tiene

Sobre cerrado de segundo precio (Vickrey)

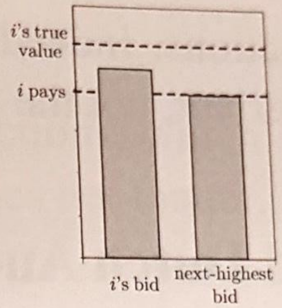
- Probablemente el tipo de subasta más intuitivo
- Procedimiento:
  1. Ronda única, en la que los ofertantes envían sus ofertas privadamente al subastador
  2. El subastador selecciona al agente con la oferta más alta
  3. El agente tiene que pagar la cantidad de la segunda oferta más alta
- Estrategia dominante: los postores deben decir la verdad
- Poco propensa a la manipulación estratégica
- No muy popular en la vida real, pero muy utilizada en sistemas de subastas computacionales
- Estrategia dominante: Decir la verdad y pujar por nuestro precio límite



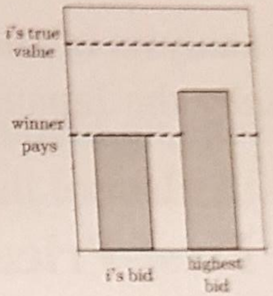
(a) Bidding honestly, i has the highest bid.



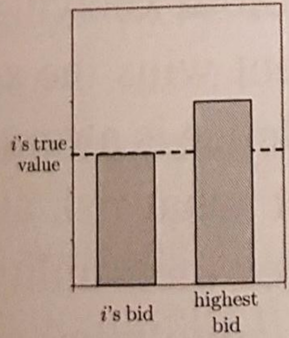
(b) i bids higher and still wins.



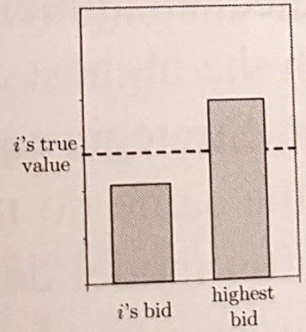
(c) i bids lower and still wins.



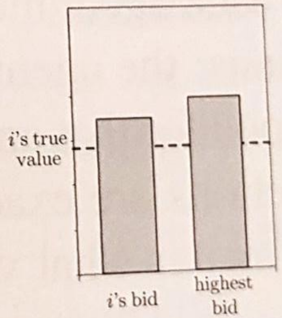
(d) i bids even lower and loses.



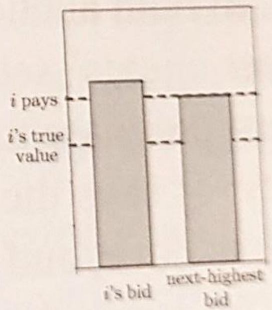
(e) Bidding honestly, i does not have the highest bid.



(f) i bids lower and still loses.



(g) i bids higher and still loses.



(h) i bids even higher and wins.

A modo de resumen

Inglesa, japonesa: first-price, open-cry, ascendente.

Holandesa: first-price, open-cry, descendente.

FPSB: first-price, sealed-bid, one-shot.

Vickrey: second-price, sealed-bid, one-shot.

Análisis: Subastas por un solo bien

Teorema de Equivalencia de Ingresos

- Siempre que se cumplan ciertas condiciones, el precio promedio al que se termina vendiendo el bien es el mismo en los tipos de subastas analizados.
- Las condiciones que deben cumplirse son:
  - Los jugadores son neutrales al riesgo.
  - Cada jugador conoce su valuación y ésta es privada e independiente.
  - Los jugadores son simétricos, es decir, que poseen la misma información y sus estimaciones sobre las valuaciones de los otros participantes son iguales.
  - El pago depende únicamente de las pujas realizadas.

Mentiras y confabulaciones (lies and collusions)

- Los protocolos anteriores pueden ser manipulados por agentes estratégicos:
  - **Subastador mentiroso**: problema para las subastas Vickrey, no para las open-cry ni la FPSB.
  - **Cómplices**: utilizados por el subastador para incrementar las subastas artificialmente (ej: inglesa).
  - **Colusión**: grupos de apostantes que cooperan para estafar al sistema:

    ➔ Inglesa: todos apuestan a la baja, no interesa romper el pacto.

    ➔ Vickrey: todos apuestan a la baja excepto el ganador, que da su valoración real.

    ➔ Menos común en la holandesa y la FPSB ya que alguien podría romper el pacto en su beneficio.

- Soluciones:
  - Inmunidad a las confabulaciones: los postores no se conocen entre sí
  - Honestidad del subastador: subastas open-cry o gestionada por terceros (especialmente en el caso de una subasta de segundo precio)

Subastas multi-unidad

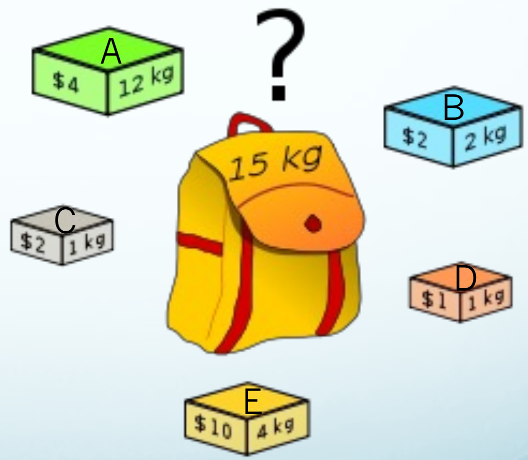
- Se dispone de varias unidades del item a subastar
- Ej: Un subastador quiere vender 15 manzanas maximizando los ingresos.
- Subasta secuencial / Subasta paralela
  - Hacer una oferta es extremadamente difícil
  - Los postores pueden terminar con menos unidades de las necesarias
- Soluciones
  - Hay formatos de unidades múltiples de los protocolos de una sola unidad
    - Inglesa, japonesa, holandesa, sobre cerrado
  - Los postores ofertan por cantidad y precio

Subastas multi-unidad

- Un subastador quiere vender 15 manzanas maximizando los ingresos.
- El subastador recibe las siguientes ofertas:
  - A: Compra 12 manzanas por 4 €
  - B: Compra 2 manzanas por 2 €
  - C: Compra 1 manzana por 2 €
  - D: Compra 1 manzana por 1 €
  - E: Compra 4 manzanas por 10 €
- ¿Cuál es la asignación óptima?

Subastas multi-unidad

- Un subastador quiere vender 15 manzanas maximizando los ingresos.
- El subastador recibe las siguientes ofertas:
  - A: Compra 12 manzanas por 4 €
  - B: Compra 2 manzanas por 2 €
  - C: Compra 1 manzana por 2 €
  - D: Compra 1 manzana por 1 €
  - E: Compra 4 manzanas por 10 €



Problema de la mochila

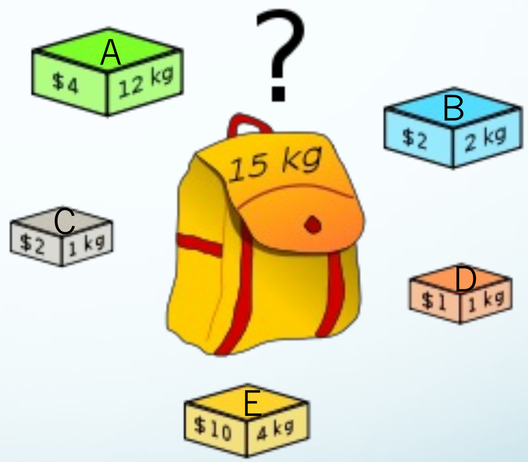
- Una variable binaria para cada oferta: x_A, x_B, x_C, x_D, x_E
- Maximizar los ingresos obtenidos al llenar la mochila

Maximizar $\quad 4x_A + 2x_B + 2x_C + x_D + 10x_E$

Sujeto a

$$12x_A + 2x_B + x_C + x_D + 4x_E \leq 15$$

$$x_A, x_B, x_C, x_D, x_E \in \{0,1\}$$

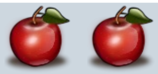


Las subastas multi-unidad permiten ofertas más complejas

- Un subastador quiere vender 15 manzanas maximizando los ingresos.
- El subastador recibe las siguientes ofertas:
  - A: Compra 12 manzanas por 4 € OR 6 manzanas por 3 €
  - B: Compra 2 manzanas por 2 €
  - C: Compra 1 manzana por 2 €
  - D: Compra 1 manzana por 1 €
  - E: Compra 4 manzanas por 10 € OR 2 manzanas por 6 €
- ¿Cuál es la asignación óptima?

Lenguajes de ofertas

- En general, en una subasta de múltiples unidades, los participantes deben especificar su valoración para cada número de unidades.
- Si estamos vendiendo tres manzanas, cada postor debe decirnos algo así como

	**4€**
	6€
	7€

- Imaginad que estamos vendiendo 100 manzanas …
- Los lenguajes de ofertas permiten a los ofertantes transmitir esta información de forma más compacta

Lenguajes de ofertas

- Una aproximación son la ofertas XOR

$$\beta_i = (\{a,b\}, 3)\ XOR\ (\{c,d\}, 5)$$

- XOR porque pagaremos por al menos una.
- El significado de la oferta es:

  Pagaría 3 por un lote que contenga *a* y *b* pero no c y *d*. Pagaré 5 por un lote que contenga c y *d* de pero no *a* y *b*, y pagaré 5 por un lote que contenga *a*, *b*, *c* y *d*.
- A partir de esto podemos construir una valoración.

$$u_{\beta 1}(\{a\}) = 0,\ u_{\beta 1}(\{b\}) = 0,\ u_{\beta 1}(\{a, b\}) = 3,\ u_{\beta 1}(\{c, d\}) = 5,$$
$$u_{\beta 1}(\{a, b, c, d\}) = 5$$

Lenguajes de ofertas

- Más formalmente, una oferta como esta:

$$\beta = (g_1, p_1)\ XOR\ \ldots\ldots..\ XOR\ (g_k, p_k)$$

define la siguiente valoración v_β:

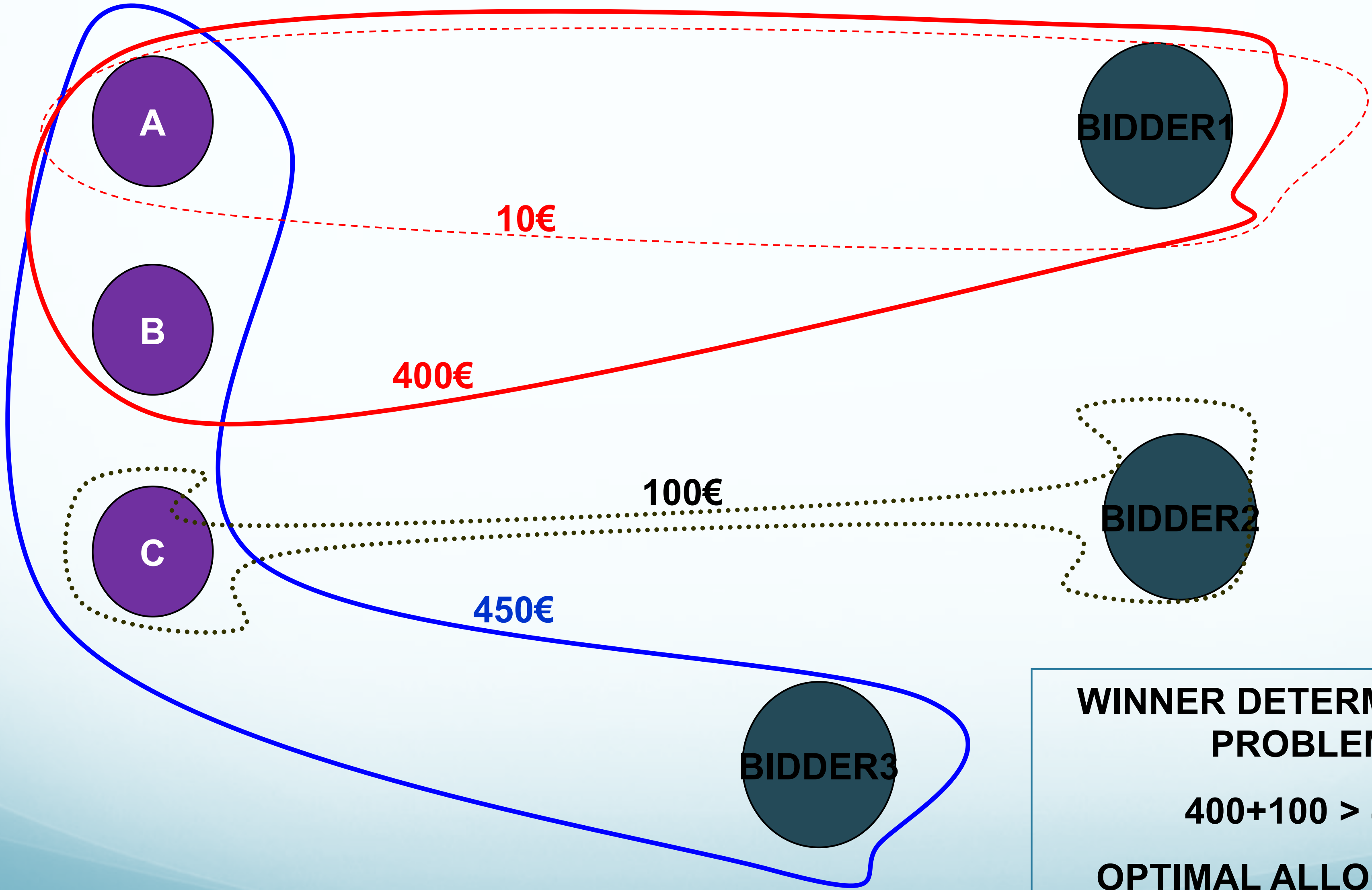
$$u_\beta(g') = \begin{cases} 0\ si\ g'\ no\ satisface\ ninguna\ oferta\ (g_i, p_i) \\ \max\{p_i \mid g_i \subseteq g'\}\ en\ cualquier\ otro\ caso \end{cases}$$

- Las ofertas XOR son totalmente expresivas, es decir pueden expresar cualquier función de valoración sobre un conjunto de mercancías.
- Para ello, es posible que necesitemos un número exponencial de ofertas atómicas.
- Sin embargo, la valoración de un lote se puede calcular en tiempo polinómico.

Subastas combinatorias

- En las subastas combinatorias, diferentes bienes se comercializan simultáneamente.
- Es importante cuando las valoraciones de los postores dependen en gran medida del conjunto de bienes que reciben.
- Ejemplos:
  - Subastas de energía
  - Subastas de compras corporativas
  - Subastas de rutas en una red (es decir, conexiones de comunicación)

Subastas combinatorias



WINNER DETERMINATION PROBLEM:

400+100 > 450

OPTIMAL ALLOCATION: Bidders 1 and 2 win

Subastas combinatorias

- Un cjto. de postores $N = \{1, \ldots, n\}$
- Un cjto. de bienes $G = \{1, \ldots, m\}$
- Para cada postor $i \in N$, disponemos de una función v_i

$$v_i : P(G) \to \mathbb{R}$$

- Interesante cuando los postores tienen funciones de valoración no aditiva:
  - Sustituibilidad  
  - Complementariedad 
- Subastas secuenciales / paralelas : The exposure problem.

Subastas combinatorias

- El resultado de una subasta combinatoria es una asignación de los bienes a los postores (agentes).

¿Quién gana entonces?

- El problema de la determinación del ganador (**WDP** – Winner Determination Problem) supone:
  - *Encontrar el conjunto de apuestas ganadoras que maximizan los ingresos del subastador*.
- El WDP es un problema NP-completo.
- ¿Cuál es la solución optima de esta subasta combinatoria?
  - 3 agentes Ag1, Ag2, Ag3 y 4 ítems a subastar (a,b,c,d)
  - Agente 1: ({a,b}, 5), ({b,c}, 7), ({c,d}, 6)
  - Agente 2: ({a,d}, 7), ({a,c,d}, 8)
  - Agente 3: ({b}, 5), ({a,b,c,d}, 12)

Subastas combinatorias

- Formalmente una asignación es una lista de conjuntos de g_1, ….., g_n, uno para cada agente Ag_i con la condición de que

$$g_i \subseteq G$$

y que $\forall i,j \in Ag$, tal que $i \neq j$, se cumple que $g_i \cap g_j = \emptyset$

- No se asigna un mismo bien (producto) a más de un agente.

- El conjunto de todas las asignaciones G a los agentes Ag es:

$$alloc\ (G, Ag)$$

Subastas combinatorias

- Cuando diseñamos la subasta, tenemos que decir cómo se determina la asignación.
- ¿Cómo se haría para **asignación de tareas**?
- Una forma natural es maximizar el bienestar social
  - Suma de las utilidades de todos los agentes
- Definimos una función de bienestar social:

$$F_N(g_1,\ldots\ldots,g_n,u_1,\ldots\ldots\ldots,u_n)=\sum_{i=1}^{n}u_i(g_i)$$

Subastas combinatorias

- Teniendo en cuenta, esto podemos definir una subasta combinatoria.

- Dado un conjunto de bienes (tareas) *G* y un conjunto de funciones de valuación (utilidad) $u_1, \ldots\ldots, u_n$ una por cada agente $i \in Ag$, el objetivo es encontrar una asignación

$$g_1^*, g_2^*, \ldots\ldots\ldots\ldots\ldots, g_n^*$$

que maximice F_N, es decir:

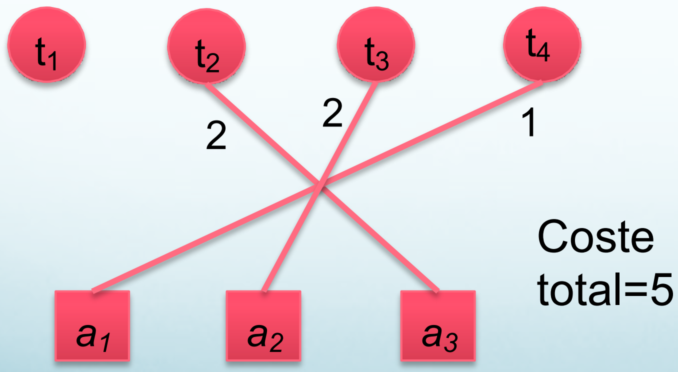
$$g_1^*, g_2^*, \ldots\ldots\ldots\ldots\ldots, g_n^* = \underset{(g_1, g_2, \ldots\ldots, g_n) \in alloc(G, Ag)}{\arg\max} F_N(g_1, g_2, \ldots\ldots, g_n, u_1, u_2, \ldots\ldots, u_n)$$

Calcular esto es determinar el ganador

Un ejemplo: Alg. Bertsekas

Aplicado a un problema de asignación (**task allocation**):

- *n* agentes y *m* tareas (m>=*n*)
- Cada tarea *t* puede ser asignada a cualquier agente *a* con un coste: *c(t,a)*
- Cada tarea sólo puede ser asignada a un agente y cada agente recibe exactamente una tarea
- Objetivo: **Minimizar el coste total de la asignación global**



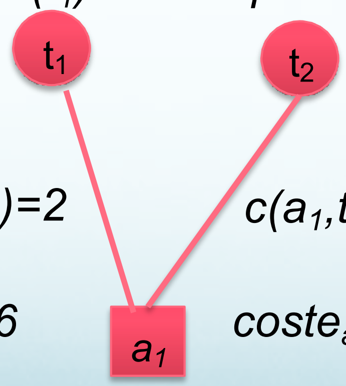
C(t,a)	t_1	t_2	t_3	t_4
a_1	*2*	*1*	*4*	*1*
a_2	*4*	*7*	*2*	*8*
a_3	*6*	*2*	*1*	*9*

Algoritmo de subasta de Bertsekas

Idea general:

- Agentes pujan por tareas (la menos costosa)
- Las tareas tienen un precio
- El coste real de una tarea es la suma del coste que estima el agente + el precio que tiene la tarea
- El valor de la puja es la diferencia entre el segundo menor coste y el menor coste
- Si hay varios agentes que quieren la misma tarea, ésta aumenta su precio

precio(t_1)=4 *precio(t_2)=1*



$c(a_1,t_1)=2$ $c(a_1,t_2)=3$

$coste_{a1}(t_1)=2+4=6$ $coste_{a1}(t_2)=3+1=4$

a_1 puja por t_2 con $puja_{a1}=6-4=2$

El valor de la puja representa la ventaja de la tarea t_2 respecto a la siguiente mejor (t_1).
Si alguien pagase más por t_2, el agente cambiaría su opinión, pujando por t_1.

Algoritmo de subasta de Bertsekas

Inicialización:

- *AS:={};* $\forall t_i$: *precio*(t_i)*:=0;*

Repetir

Fase de apuestas:

- Para cada agente a_i no asignado:
  - Calcula el coste para cada tarea: $cost_{ai}(t_j)=c(a_i,t_j)+precio(t_j)$
  - Sean t_k y t_s las tareas con el menor y segundo menor coste de todas
  - Emite una puja a la tarea de menor coste (t_k) con valor:

$$puja_{ai}=cost_{ai}(t_s)-cost_{ai}(t_k)$$

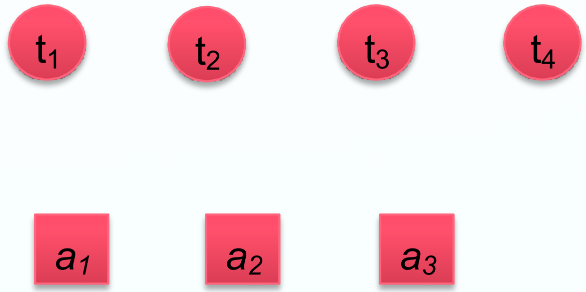
Fase de asignación:

- Para cada tarea t_j que ha recibido por lo menos una puja:
  - Sea a_u el agente con la puja ($puja_{au}$) más alta para t_j
  - $AS:= (AS/\{<\_,t_j>\})\cup\{<a_u,t_j>\}$ // asigna t_j a a_u eliminando la asignación anterior
  - $precio(t_j):=precio(t_j)+puja_{au}$

Hasta que todas las tareas/agentes esten asignadas/os

Algoritmo de subasta de Bertsekas

Ejemplo:

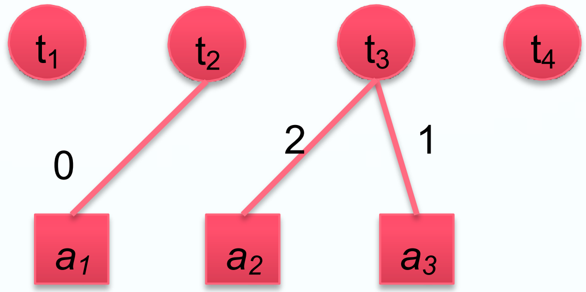


C(t,a)	t_1	t_2	t_3	t_4
a_1	*2*	*1*	*4*	*1*
a_2	*4*	*7*	*2*	*8*
a_3	*6*	*2*	*1*	*9*

Iteración		t_1	t_2	t_3	t_4
0	precio	*0*	*0*	*0*	*0*
	pujas				
	asignación				
	precio				
	pujas				
	asignación				
	precio				
	pujas				
	asignación				
	precio				

Asignación óptima:
Algoritmo de subasta de Bertsekas

Ejemplo:

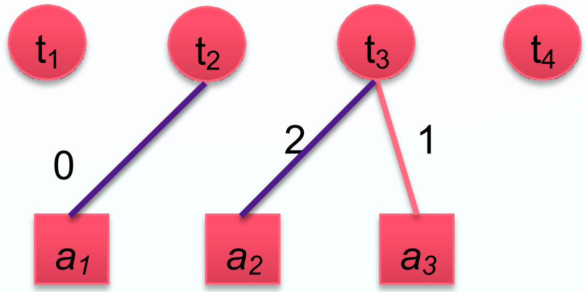


C(t,a)	***t_1***	***t_2***	***t_3***	***t_4***
a_1	*2*	*1*	*4*	*1*
a_2	*4*	*7*	*2*	*8*
a_3	*6*	*2*	*1*	*9*

Iteración		t_1	t_2	t_3	t_4
0	precio	*0*	*0*	*0*	*0*
	pujas		$a_1(0)$	$a_2(2)$; $a_3(1)$	
	asignación				
	precio				
	pujas				
	asignación				
	precio				
	pujas				
	asignación				
	precio				

Asignación óptima:
Algoritmo de subasta de Bertsekas

Ejemplo:

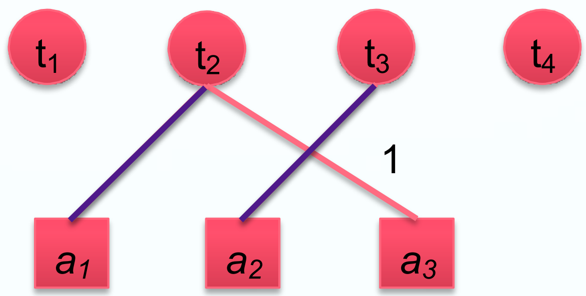


C(t,a)	***t_1***	***t_2***	***t_3***	***t_4***
a_1	*2*	*1*	*4*	*1*
a_2	*4*	*7*	*2*	*8*
a_3	*6*	*2*	*1*	*9*

Iteración		*t_1*	*t_2*	*t_3*	*t_4*
0	precio	*0*	*0*	*0*	*0*
	pujas		*$a_1(0)$*	*$a_2(2)$; $a_3(1)$*	
	asignación		*a_1*	*a_2*	
	precio	*0*	*0*	*2*	*0*
	pujas				
	asignación				
	precio				
	pujas				
	asignación				
	precio				

Asignación óptima:
Algoritmo de subasta de Bertsekas

Ejemplo:

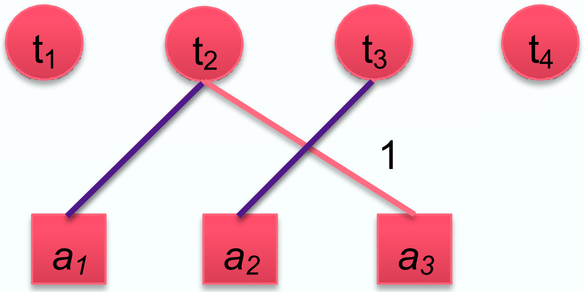


C(t,a)	*t_1*	*t_2*	*t_3*	*t_4*
a_1	*2*	*1*	*4*	*1*
a_2	*4*	*7*	*2*	*8*
a_3	*6*	*2*	*1*	*9*

Iteración		t_1	t_2	t_3	t_4
0	precio	*0*	*0*	*0*	*0*
	pujas		$a_1(0)$	$a_2(2)$; $a_3(1)$	
	asignación		a_1	a_2	
	precio	*0*	*0*	*2*	*0*
	pujas		$a_3(1)$		
	asignación				
	precio				
	pujas				
	asignación				
	precio				

Asignación óptima:
Algoritmo de subasta de Bertsekas

Ejemplo:



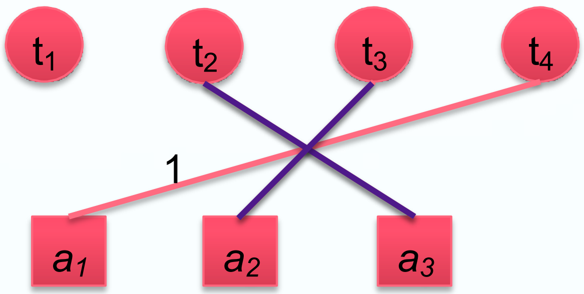
C(t,a)	t_1	t_2	t_3	t_4
a_1	*2*	*1*	*4*	*1*
a_2	*4*	*7*	*2*	*8*
a_3	*6*	*2*	*1*	*9*

Iteración		t_1	t_2	t_3	t_4
0	precio	*0*	*0*	*0*	*0*
	pujas		$a_1(0)$	$a_2(2)$; $a_3(1)$	
	asignación		a_1	a_2	
	precio	*0*	*0*	*2*	*0*
	pujas		$a_3(1)$		
	asignación		a_3	a_2	
	precio	*0*	*1*	*2*	*0*
	pujas				
	asignación				
	precio				

Asignación óptima:

Algoritmo de subasta de Bertsekas

Ejemplo:



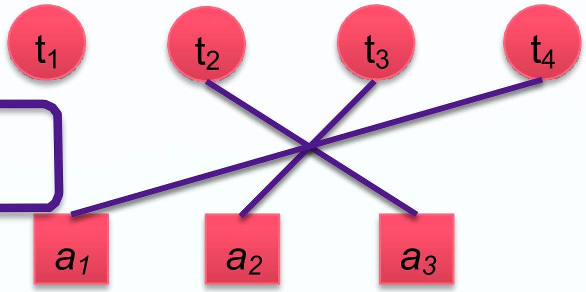
C(t,a)	t_1	t_2	t_3	t_4
a_1	*2*	*1*	*4*	*1*
a_2	*4*	*7*	*2*	*8*
a_3	*6*	*2*	*1*	*9*

Iteración		t_1	t_2	t_3	t_4
0	precio	*0*	*0*	*0*	*0*
	pujas		$a_1(0)$	$a_2(2)$; $a_3(1)$	
	asignación		a_1	a_2	
	precio	*0*	*0*	*2*	*0*
	pujas		$a_3(1)$		
	asignación		a_3	a_2	
	precio	*0*	*1*	*2*	*0*
	pujas				$a_1(1)$
	asignación				
	precio				

Asignación óptima:
Algoritmo de subasta de Bertsekas

Ejemplo:

coste total=5



C(t,a)	t_1	t_2	t_3	t_4
a_1	2	1	4	1
a_2	4	7	2	8
a_3	6	2	1	9

Iteración		t_1	t_2	t_3	t_4
0	precio	0	0	0	0
1	pujas		$a_1(0)$	$a_2(2)$; $a_3(1)$	
	asignación		a_1	a_2	
	precio	0	0	2	0
2	pujas		$a_3(1)$		
	asignación		a_3	a_2	
	precio	0	1	2	0
3	pujas				$a_1(1)$
	asignación		a_3	a_2	a_1
	precio	0	1	2	1

Algoritmo de subasta de Bertsekas

- Se añade ε a los precios para evitar iteraciones infinitas en algunos casos
- El algoritmo encuentra la solución óptima si:
  - los costes son enteros y $\varepsilon < 1/n$

- Si m<n, entonces tareas pujan por agentes
- Complejidad (peor caso): $O(n^3C/\varepsilon)$, donde C es el máximo coste
- Se puede reducir la complejidad a $O(n^3 log(nC))$ con algunas técnicas sobre ε

La aplicación en la asignación de SimFleet es directa:

- Agentes ➔➔ Taxis
- Tareas ➔➔ Clientes
- Coste ➔➔ por ejemplo distancia/tiempo esperado de trayecto