Università degli Studi di Verona

Teoria Computazionale dei Giochi

Set di Esercizi

Mattia Zorzan

27 giugno 2021

Indice

La presente dispensa LaTeX presenta le soluzioni per i *6 Set* di esercizi assegnatici dal Prof. Ferdinando Cicalese durante il corso **Teoria Computazionale dei Giochi** per la Laurea Magistrale in **Ingegneria e Scienze Informatiche**, A.A. 2020/2021.
I Set di esercizi potrebbero cambiare negli A.A. a seguire ed eventuali nuove pubblicazioni potrebbero smentire alcuni risultati portati nella presente. Le consegne non sono inoltre riportate per intero, quindi anche se simili potrebbero richiedere cose diverse da quelle svolte.

1 Set 1 - Giochi classici, Equilibri e Aste

1.1 Esercizio 1

Si dimostri che il gioco "Morra Cinese" ammette un solo MNE

Definiamo gli outcome del gioco in forma matriciale

			P_2	
		R	P	S
	R	$(0,0)$	$(-1,1)$	$(1,-1)$
P_1	P	$(1,-1)$	$(0,0)$	$(-1,1)$
	S	$(-1,1)$	$(1,-1)$	$(0,0)$

È facile vedere come, fissando la strategia dell'altro giocatore, esista sempre una *improving move* che migiori la situazione di un giocatore.
Non esiste quindi un PNE, devo utilizzare un equilibrio di tipo misto, randomizzando le strategie, detto MNE.
Supponiamo una distribuzione uniforme di probabilità, quindi $P_1 < \frac{1}{3}, \frac{1}{3}, \frac{1}{3} >$ e $P_2 < \frac{1}{3}, \frac{1}{3}, \frac{1}{3} >$, in questo caso l'utilità di ogni risposta è 0

$$u_i = \frac{1}{3} \cdot \frac{1}{3} \cdot u(R, R) \cdot 6 = 0$$

Si noti che u_i sarà sempre in questa forma per questo tipo di distribuzione, essedo *Morra Cinese* un **TWO PLAYERS ZERO-SUM GAME**.
Prendiamo ora una differente distribuzione $P_1 < \frac{2}{3}, 0, \frac{1}{3} >$ e $P_2 < \frac{1}{3}, \frac{2}{3}, 0 >$, in questo caso

$$\begin{aligned} u_i &= \frac{2}{3} \cdot \frac{1}{3} \cdot u(R, R) + \frac{2}{3} \cdot \frac{2}{3} \cdot u(R, P) + \frac{1}{3} \cdot \frac{1}{3} \cdot u(S, R) + \frac{1}{3} \cdot \frac{2}{3} \cdot u(S, P) \\ &= 0 - \frac{4}{9} - \frac{2}{9} + \frac{1}{3} = -\frac{1}{3} \end{aligned}$$

abbiamo quindi utilità negativa. È facile vedere, applicando la stessa tecnica e risolvendo la corrispondente equazione, come tutte le altre distribuzioni diano utilità negativa.
Quindi la distribuzione uniforme $\frac{1}{3}$ per entrambi i giocatori è l'unico MNE possibile.

1.2 Esercizio 2

Si dimostri che il seguente gioco ammette due PNE

		P_2	
		A	B
P_1	A	$(3,5)$	$(2,2)$
	B	$(1,1)$	$(4,6)$

Si possono individuare i due seguenti PNE:

- $\text{PNE}_1 = (\text{B,B})$ Dato che le utilità per entrmbi i giocatori sono massime per ogni possibile outcome.
- $\text{PNE}_2 = (\text{A,A})$ Dato che fissata la mossa per l'altro giocatore, non esiste una imporving move che migliori la propria situazione

Nello psecifico, PNE_1 è raggiungibile se la prima mossa (sia essa di P_1 o P_2) è B, PNE_2 se invece è A

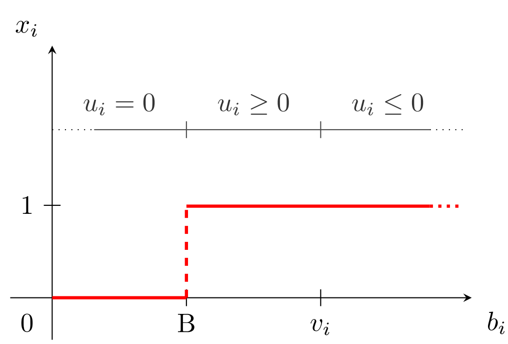
1.3 Esercizio 3

Si consideri la First Price Auction con n partecipanti, si dimostri se essa è DSIC o meno

Analizziamo i possibili outcome, data la valutazione v_i, $\forall i \in [n]$ dove n è il numero dei partecipanti, distinguiamo due possibili profili d'offerta:

- $b_i \geq v_i$: Che risulterà in $u_i \leq 0$, dato che nel caso migliore (quando $x_i = 1$) avrò utilità 0, negativa in ogni altro caso
- $b_i < v_i$: L'unico caso in cui $u_i > 0$, a patto di allocare

Possiamo quindi dire che l'unico incentivo che un partecipante ha per aumentare la propria utilità è *sotto-offrire*. Non avendo incentivo nell'offrire sinceramente, questo modello non è DSIC.
Sia $B = \max_{j \neq i} b_j$



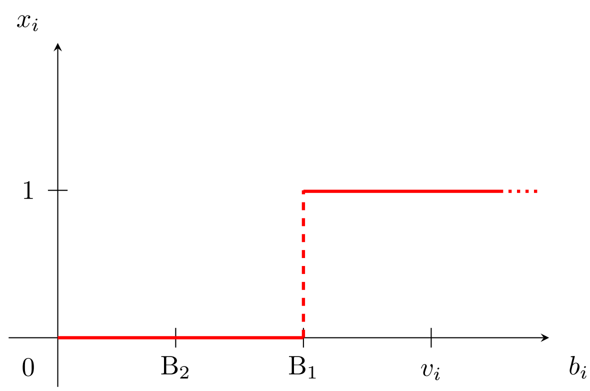
1.4 Esercizio 4

Si consideri una Third Price Auction con $n \geq 3$ partecipanti, si dimostri se essa è DISC o meno

Definiamo

- $\mathrm{B}_1 = \max_{j \neq i} b_j$
- $\mathrm{B}_2 = \max_{k \neq i, k \neq j} b_k$

come la penultima e terzultima offerta più alta, quest'ultima fisserà il prezzo.



Essendo $u_i = v_i - p$ se il partecipante i alloca, 0 altrimenti, distinguiamo 3 scenari

- $v_i > \mathrm{B}_1$: In questo caso, essendo fissati v_i e B_2 (ovvero il prezzo) avrò sempre $u_i \geq 0$ e costante da B_1 in poi. Non ho quindi alcun incentivo nell'offrire non sinceramente.
- $v_i < \mathrm{B}_2$: Ho sempre utilità negativa
- $\mathrm{B}_2 \leq v_i \leq \mathrm{B}_1$: In questo caso ho $u_i \geq 0$ ma, offrendo sinceramente, non alloco. Mi è quindi conveniente *sovra-offrire* al fine di avre effettivamente $u_i \geq 0$, avrei $u_i = 0$ altrimenti.

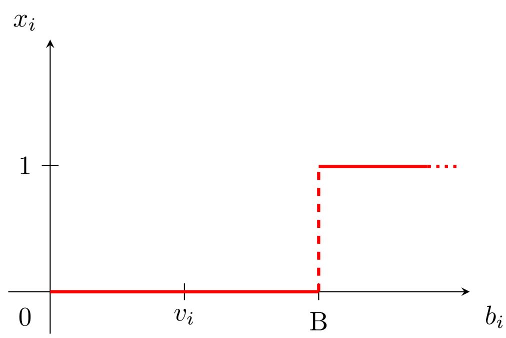
In questo caso ho incentivo nell'offrire $b_i \neq v_i$, al fine di allocare. Per questo motivo il meccanismo proposto non è DSIC.

1.5 Esercizio 5

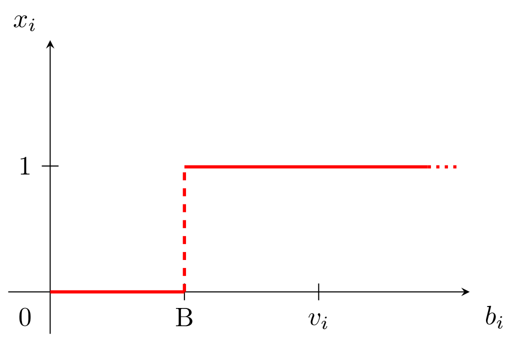
Si dimostri che la Vickrey Auction è DSIC

Sia B = $\max_{j \neq i} b_j$, si possono distinguere due scenari

- $v_i \leq B$: Dato che per allocare (ed avere quindi $u_i \neq 0$) devo necessariamente presentare $b_i \geq B$, $v_i = \max\{0, v_i - B\} = 0$



- $v_i > B$: Presentando una qualsiasi offerta $b_i > B$ alloco ed ottengo utilità $u_i = max\{0, v_i - B\} = v_i - B$. Dato che sia v_i che B sono valori fissati, avrò utlità costante e quindi nessun incentivo nell'offire $b_i \neq v_i$



Da questo, la Vickrey Auction è DSIC.

1.6 Esercizio 6

Si generalizzi la Vickrey Auction per bandire k oggetti a $n > k$ partecipanti, ogni $i \in [n]$ può allocare al più un oggetto. Date le regole di allocazione e prezzo, si dimostri che il nuovo meccanismo è DSIC

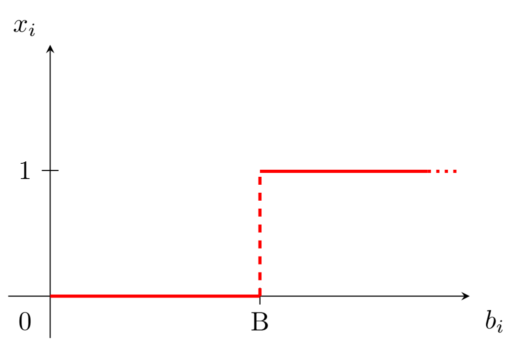
Data la funzione di allocazione

$$x_i(b_i) = \begin{cases} 1 & \text{se } n \neq 0 \wedge b_i > B \\ 0 & \text{altrimenti} \end{cases}$$

stabiliamo un ordinamento su tutte le offerte. Chiamo **B** l'insieme ordinato delle offerte

$$\mathbf{B} = \{b_i \mid i \in [n]\}$$

tale che $b_1 \geq b_2 \geq \cdots \geq b_n$
Se dopo ogni allocazione rimuoviamo da **B** il partecipante che alloca, ri-ettichettando successivamente tutti i restanti, la funzione di allocazione $x_i(b_i)$ è, $\forall i \in [n]$, monotona crescente nella forma



dove $\mathrm{B} = \max_{j \neq i} b_j$.
Definisco ora la regola di prezzo come

$$p(x_i) = \begin{cases} \mathrm{B} & \text{se } x_i = 1 \\ 0 & \text{altrimenti} \end{cases}$$

Essendo questo meccanismo una generalizzazione della Vickrey Auction, posso ripetere (come anticipato prima) il sistema di allocazione e pagamento fino a finire gli oggetti.
Essendo la funzione di allocazione monotona non decrescente ed esistendo una regola di prezzo, valendo $\forall i \in [n]$ le stesse casistiche di utlità dimostrate nell'esercizio precedente, il meccanismo è DSIC e implementabile secondo il *Mayerson's Lemma.*

2 Set 2 - Mechanism Design

2.1 Esercizio 1

Si definisca una variazione della Vickrey Auction tale per cui sia garantito il un ritorno al venditore di un costo c, che rimanga DSIC e sia Welfare Maximizing

Definiamo la situazione di partenza, il nostro meccanismo avrà n partecipanti, dove ogni $i \in [n]$ ha una valutazione *privata* v_i.
Definisco il *Social Welfare* come

$$\begin{aligned} sw &= \sum_{i=1}^{n} x_i \cdot v_i - c \\ &= v_i - c && \text{(essendo asta a singolo oggetto)} \end{aligned}$$

ovvero una piccola variazione della definizione vista a lezione, questa infatti introduce la sottrazione di un costo fisso c che dovremo garantire per struttura del meccanismo.
Dovrò procedere in due passi per lo svolgimento dell'esercizio, utilizzo prima il *Meyerson's Lemma* per dimostrare che il meccanismo è DSIC e, quindi, implementabile per poi dimostrare il fatto che è Welfare Maximizing.
Definiamo la regola d'allocazione come segue, sia $B = \max_{j \neq i} b_j$

$$x_i(b_i) = \begin{cases} 1 & \text{se } b_i \geq \max\{c, B\} \\ 0 & \text{altrimenti} \end{cases}$$

assumendo valutazioni sincere (cosa che posso fare dato che sto generalizzando la Vickrey Auction) si può facilmente vedere come la funzione d'allocazione sia monotona non decrescente.
Definisco ora la regola di prezzo come

$$p_i(x_i) = \begin{cases} \max\{c, B\} & \text{se } x_i = 1 \\ 0 & \text{altrimenti} \end{cases}$$

e, da questo, il meccanismo è DSIC e implementabile secondo *Mayerson* e garantisce il ritorno di c.
Devo ora dimostrare che è *Welfare Maximizing*, avendo la garanzia che $b_i = v_i$, distinguiamo due scenari

- Se $v_i \geq c$, allora $sw = v_i - c$ è sempre positivo
- Se $v_i < c$, allora $sw = 0$ dato che per allocare devo avere $b_i = v_i \geq c$

2.2 Esercizio 2

Si consideri un'asta "al ribasso" dove ogni partecipante fornisce un preventivo π_i e si costruisca il meccanismo corrispondente in modo che questo sia DSIC, che allochi all'offerta più bassa e che il pagamento sia pari almeno al preventivo fornito

In quest'asta avrò dei costi stimati c_i, equivalenti alle valutazioni v_i nelle Vickrey Auctions, a loro volta *privati.*
Definita l'offerta del partecipante i come π_i, chiamo $i^* = \text{argmin}_i\{\pi_i\}$.
Definisco la regola di allocazione come

$$x_i(\pi_i) = \begin{cases} 1 & \text{se } i = i^* \\ 0 & \text{altrimenti} \end{cases}$$

e la regola di prezzo come

$$p_i(x_i) = \begin{cases} P & \text{se } x_i = 1 \\ 0 & \text{altrimenti} \end{cases}$$

dove $P = \min_{j \neq i} \pi_j$.
Allochiamo quindi al partecipante che fornisce il preventivo minore e garantiamo un pagamento almeno pari a questo, devo ora dimostrare che è DSIC.
Distinguiamo due casi:

- Se $c_i \leq P$, allora, poichè P e c_i sono fissati ed indipendenti da π_i, non ho alcun incentivo nell'offrire $\pi_i \neq c_i$, essendo l'utilità costante
- Se $c_i > P$, non alloco offrendo sinceramente, quindi $u_i = 0$. Il fatto che il pagamento sia alla seconda offerta più bassa funge da incentivo per i partecipanti, dato che sotto-offrendo si abbasserebbe sicuramente l'utilità globale.

Offrire sinceramente è quindi strategia dominante per il meccanismo, che si dimostra DSIC.

2.3 Esercizio 3

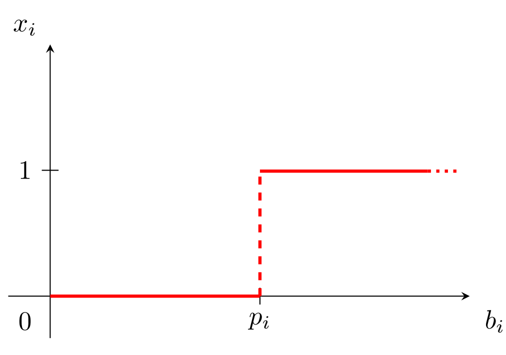
Si consideri un asta online a n partecipanti dove gli offerenti si presentano a turno, se l'oggetto non è ancora stato allocato il banditore sceglie un prezzo p_i e se $b_i \geq p_i$ alloco e termino l'asta, altrimenti i lascia l'asta e non può offrire nuovamente.
Si provi che:

- *Il meccanismo è DSIC*
- *Assumendo offerte sicere, $\forall c > 0 : sw \geq c \cdot \max_i v_i$ senza alcuna correlazione con l'ordine di arrivo (deve quindi valere per ogni profilo di prezzi)*
- *Assumendo offerte sincere, $\exists c > 0 : sw \geq c \cdot \max_i v_i$ se i pratecipanti arrivano in ordine di valutazione*

Formalizziamo la regola di allocazione come

$$x_i(b_i) = \begin{cases} 1 & \text{se } b_i \geq p_i \\ 0 & \text{altrimenti} \end{cases}$$

e, se $x_i = 1$ il partecipante i pagherà b_i (definisco così la regola di prezzo).
È facile vedere come x_i sia monotona



Dimostro ora che è DSIC:

- Se $v_i \leq p_i$, essendo $u_i = \max\{0, v_i - p_i\} = 0$, non vi è quindi incentivo nell'essere non sincero
- Se $v_i > p_i$, alloco sicuramente con $v_i = p_i$ e $u_i = \max\{0, v_i - p_i\} = c$ costante

Dimostriamo che, $\forall c > 0$, non esiste un'asta come quella appena definita t.c. si garantito

$$sw \geq c \cdot \max_i v_i$$

Data la definizione di *Social Welfare*

$$sw = \sum_{i=1}^{n} v_i \cdot x_i$$

Sostituendo otteniamo

$$\sum_{i=1}^{n} v_i \cdot x_i \geq c \cdot \max_i v_i$$

e dovrò dimostrare che vale la negazione

$$\sum_{i=1}^{n} v_i \cdot x_i < c \cdot \max_i v_i$$

Chiamo $\hat{i}$ l'unico partecipante che alloca, quindi

$$v_{\hat{i}} < c \cdot \max_i v_i$$

Ora, tramite algebra, ottengo

$$\frac{v_{\hat{i}}}{c} < \max_i v_i$$

Andiamo per casi

- Se $c > 1$ allora la disequazione è banalmente vera
- Se $c = 1$ otteniamo $v_{\hat{i}} < \max_i v_i$.
  Quello che devo dimostrare è che, per ogni scelta dei prezzi p_i esiste un profilo di valutazioni v_i t.c. valga la disequazione.
  Fissiamo i p_i, segue immediatamente che il profilo di valutazioni che rende vera la disequazione è quello dove l'agente i con la valutazione massima non alloca.

Completiamo ora l'esercizio, fissiamo il profilo di prezzo in modo che i primi $\frac{n}{2}$ siano posti ad un valore inarrivabile per le valutazioni. Estraggo quindi $V = \max\{v_1, \ldots, v_{\frac{n}{2}}\}$ e pongo i prezzi da $p_{\frac{n}{2}+1}$ a p_n a quel valore.
Garantisco in questo modo che, supponendo un profilo di valutazioni t.c. il secondo massimo (chiamato ad esempio V_2) sia nella prima metà mentre V sia nella seconda, io possa estrarre il massimo globale.
La probabilità che questo accada è $\frac{1}{4}$, che è la mia probabilità garantita di ottenere welfare.

2.4 Esercizio 4

Si consideri un'asta a k oggetti identici in cui il venditore vuole garantito un certo ritorno R. Prendiamo l'algoritmo **TARGET-REVENUE** *per stabilire le allocazioni ed i pagamenti. Si descriva formalmente la regola di allocazione, si dimostri tramite Mayerson's Lemma che il meccanismo è DSIC.*
Si dimostri inoltre che se nella Uniform Price Auction si ottene ritorno R allora la si ottiene anche in questo modello. Infine, si definisca un profilo di valutazioni tali per cui in questo modello si ottiene ritorno R ma se ne ottiene uno inferiore nella Uniform Price Auction.

Dato un profilo $B = (b_1, b_2, \dots, b_n)$ con $n > k$ e $b_1 \geq b_2 \geq \dots \geq b_n$.
La regola di allocazione è definita come

$$x_i(b_i) = \begin{cases} 1 & \text{se } i \in S^* \\ 0 & \text{altrimenti} \end{cases}$$

dove $S^* = \{i \in S \mid b_i \geq \frac{R}{|S|}\}$ e $S = \{i \mid i \in [k]\}$, R è il ritorno garantito.
Essendo x_i monotona non decrescente, definisco la regola di prezzo che la implementa

$$p_i(x_i) = \begin{cases} \frac{R}{|S|} & \text{se } x_i = 1 \\ 0 & \text{altrimenti} \end{cases}$$

DSIC per il Mayerson's Lemma.
In *Uniform Price Auction* otterò, complessivamente, $R = k \cdot (b_{k+1})$ mentre nel meccanismo appena definito otterrò $|S| \cdot (\frac{R}{|S|}) = R$, quindi allocando tutti gli oggetti entrambi i modelli d'asta mi garantiscono lo stesso ritorno R.
Il profilo per il quale vale l'ultimo statement è quello in cui ogni partecipante alloca, infatti nella *Uniform Price Auction* avremo $R = 0$ dato che non ho un'offerta $k + 1$ che non alloca, mentre nel meccanismo sopra descritto avrò ritorno R.

3 Set 3 - Strategie, Knapsack e Aste VCG

3.1 Esercizio 1

Si consideri il meccanismo d'asta definito nell'esercizio precendente. Si dimostri che il meccanismo è group-strategyproof, ovvero che non è possibile per i partecipanti accordarsi fornendo offerte non veritiere per aumentare l'utilità di almeno uno di loro
È possibile dire che un meccanismo DSIC Welfare Maximizing per un asta a k oggetti è group-strategyproof?

Per definizione, l'utilità di un partecipante i, $\forall i \in [n]$, è

$$u_i = v_i - \frac{R}{|S|}$$

Dall'algoritmo sappiamo che, siccome la valutazione v_i è data, un'offerta non sincera può solo far crescere il numero di partecipanti in S dopo l'esecuzione del ciclo *while*. Infatti presentare $b_i < v_i$ può solo far diminuire $|S|$, riducendo anche l'utlità per tutti.
Prendiamo quindi due profili d'offerta, B e B', dove

- B profilo con almeno due $b_i < v_i$, $i \in [n]$
- B profilo sincero con $b_i = v_i$, $i \in [n]$

Voglio dimostrare $u(B) \leq u(B')$.
Se i due o più partecipanti non sinceri continuano a presentare offerte b_i t.c.

$$b_i < \frac{R}{|S|}$$

allora $u(B) = u(B')$, non vi è quindi incentivo nell'offrire non sinceramente.
L'unico modo per alzare l'utilità è abbassare il prezzo e perchè questo sia possibile devo massimizzare $|S|$. Ipotizziamo quindi che b_k e b_{k+1} si accordino per offrire non sinceramente per far entrare $k+1$ in S, abbassando il prezzo e aumentando u_{k+1} da 0 a positivo.
Dato che ho solo k posti in S, l'ingresso di $k+1$ fa uscire un altro partecipante j riducendo u_j a 0. j potrà, esattamente come ha fatto $k+1$, accordarsi con il nuovo k per rientrare ripetendo essattamente la stassa procedura. Non è quindi possibile accordarsi senza ridurre l'utilità di almeno un partecipante.
Inoltre, nel modello per cui tutti pagano b_{k+1} (*Uniform Price Auction*) è sempre dominante accordarsi con i_{k+1} per fargli allocare in modo da abbassare il prezzo, alzando così l'utilità gloabale.

3.2 Esercizio 2

Si consideri una variante della Knapsack Auction con due "knapsack". Possiamo affermare che la regola di allocazione è monotona? Si assuma che ogni partecipante è compatibile con entrambi i "knapsack"

Voglio dimostrare che la regola di allocazione per la *Knapsack Auction* è monotona.
Per prima cosa impongo un ordinamento sulle offerte, avrò quindi

$$\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \cdots \geq \frac{b_n}{w_n}$$

Scelgo ora l, a differenza della normale *Knapsack Auction* ho due "knapsack" quindi avrò un l per ognuno.
Voglio saturare completamente il primo "knapsack" prima di passare al secondo, comincio quindi con S_1

$$l_1 = \max\{i \mid w_1 + w_2 + \cdots + w_{l_1} \leq W_1\}$$

Estraggo ora t, ovvero il partecipante con l'offerta più alta, come

$$t_1 = \operatorname*{argmax}_i b_i$$

Devo ora definire la regola di allocazione, distinguo due casi

- Se $b_1 + b_2 + \cdots + b_{l_1} \geq b_{t_1}$ allora la mia regola di allocazione sarà la seguente

  $$x_i^{(1)} = \begin{cases} 1 & \text{se } i \leq l_1 \\ 0 & \text{altrimenti} \end{cases}$$

  ovvero alloco ai primi l_1 partecipanti

- Se $b_1 + b_2 + \cdots + b_{l_1} < b_{t_1}$ allora

  $$x_i^{(1)} = \begin{cases} 1 & \text{se } i = t_1 \\ 0 & \text{altrimenti} \end{cases}$$

  ovvero alloco solo al partecipante t_1

Ripeto lo stesso procedimento per il secondo "knapsack", consoderando però solo i partecipanti che non hanno già allocato con S_1.
È facile vedere come le funzioni di allocazione per entrambi i "knapsack" siano monotone, dato che la composizione di funzioni monotone è monotona a sua volta possiamo considerare corretta l'affermazione.

3.3 Esercizio 3

Si consideri un'asta combinatoria a 2 oggetti e 3 partecipanti, date le seguenti valutazioni

$$v_1(AB) = 1, v_1(A) = v_1(B) = v_1(\varnothing) = 0$$
$$v_2(AB) = v_2(A) = 1, v_2(B) = v_2(\varnothing) = 0$$
$$v_3(AB) = v_3(B) = 1, v_3(A) = v_3(\varnothing) = 0$$

Si calcolino assegnamento e pagamenti per VCG, si ripeta lo stesso calcolo ma solo per i primi 2 partecipanti. Si confronti infine il profitto tra i due casi

Devo prima di tutto definire la regola di allocazione, nei meccanismi VCG sappiamo essere

$$\omega^* = \underset{\omega \in \Omega}{\text{argmax}} \sum_{i=1}^{n} b_i(\omega)$$

Devo quindi cercare l'allocazione che massimizza il welfare, nel nostro caso

- i_1 non alloca nulla
- i_2 alloca A
- i_3 alloca B

I pagamenti nel meccanismo VCG sono definiti come

$$p_i(\omega^*) = \max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) - \sum_{j \neq i} b_j(\omega^*)$$

e nel nostro caso

- Per i_1
$$2 - 2 = 0$$
- Per i_2
$$1 - 1 = 0$$
- Per i_3
$$1 - 1 = 0$$

Rimuovendo i_3 ho una variazione nella mia asta, infatti posso scegliere tra

- Allocare tutto ad i_1, ottenendo welfare 1
- Allocare A ad i_2, ottenendo welfare 1

Nel primo caso avrò

- Per i_1
$$1 - 0 = 1$$

- Per i_2

$$1 - 1 = 0$$

Nel secondo caso

- Per i_1

$$1 - 1 = 0$$

- Per i_2

$$1 - 0 = 1$$

Nello scenario in cui tutti e tre i partecipanti presentano le proprie offerte avrò profitto 0, in quello in cui le offerte presentate sono quelle di due soli partecipanti avrò profitto 1.

4 Set 4 - Virtual Welfare

4.1 Esercizio 1

Si calcolino le funzioni di valutazione virtuale per le seguenti distribuzioni dicendo per ognuna di esse se è regolare

- Distribuzione $U[0, a]$ con $a > 0$

  Definiamo $F(v)$ data la distribuzione uniforme

  $$F(v) = \begin{cases} 0 & \text{se } v < 0 \\ v & \text{se } 0 \leq v \leq a \\ a & \text{se } v > a \end{cases}$$

  Ricordiamo che

  $$E_{v \sim U[0,a]}[v] = \int_{-\infty}^{+\infty} v \cdot f(v)\, dv$$

  dove $f(v)$, ovvero la nostra *densità*, è

  $$f(v) = \begin{cases} 0 & \text{se } v < 0 \\ a & \text{se } 0 \leq v \leq a \\ 0 & \text{se } v > a \end{cases}$$

  Possiamo in questo caso prendere il nostro $E_{v \sim U[0,a]}[v] = \frac{a}{2}v^2$, risolvendo l'integrale.
  Calcoliamo ora VV, la nostra funzione di valutazione virtuale

  $$\begin{aligned} \text{VV} &= \frac{a}{2}v^2 - \frac{1-a}{0} \\ &= \frac{a}{2}v^2 - \infty \end{aligned}$$

  sempre decrescente. Quindi $F = U[0, a]$ con $a > 0$ non è regolare.

- Distribuzione $F(v) = 1 - \frac{1}{(v+1)^c}$ con c > 0 costante

  Derivo $F(v)$ per ottenere $f(v)$

$$
\begin{aligned}
f(v) &= \frac{\partial}{\partial\, v}\ F(v) \\
&= -\frac{\partial}{\partial\, v}\ \frac{1}{(v-1)^c} \\
&= \frac{\partial}{\partial\, u}\ \frac{1}{u^c} \cdot \frac{\partial}{\partial\, v}\ u \text{ con } u = v - 1 \\
&= \frac{cu^{c-1}}{u^{2c}} \cdot \frac{\partial}{\partial\, v}\ u \\
&= \frac{c(v-1)^{c-1} \frac{\partial}{\partial\, v}\ (v-1)}{(v-1)^{2c}} \\
&= c(v-1)^{c+1}
\end{aligned}
$$

  ed integro per ottenere il valore atteso dalla mia distribuzione

$$
\begin{aligned}
E_{v\sim F}[v] &= \int_{-\infty}^{+\infty} cv(v-1)^{c+1}\ dv \\
&= c \cdot \int_{-\infty}^{+\infty} v(v-1)^{c+1}\ dv \\
&\simeq +\infty
\end{aligned}
$$

  Sostituendo, calcolo VV

$$
\begin{aligned}
\text{VV} &= +\infty - \frac{1 - 1 + \frac{1}{(+\infty+1)^c}}{+\infty} \\
&= +\infty - \frac{0}{+\infty} \\
&= +\infty
\end{aligned}
$$

  Ora, cos'è $+\infty$? Una costante? Una variabile? Bella domanda filosofica...
  Nel dubbio diciamo che è non decrescente, quindi regolare.

4.2 Esercizio 2

Si consideri un'asta a k oggetti identici in cui le distribuzioni F siano date in funzione della distribuzione $U[0,1]$. Si fornisca un'asta ottima che sia welfare maximizing. Da quale dei seguenti dipende il prezzo di riserva? k, n, F?

Assumiamo valutazioni sincere. Ogni partecipante i, $i \in [n]$, fornirà un'offerta b_i, in base alla quale calcoleremo la sua valutazione virtuale.
Dato che tutti utilizzano la distribuzione uniforme, tutti avranno la medesima $F(v)$

$$F(v) = \begin{cases} 0 & \text{se } v < 0 \\ v & \text{se } 0 \leq v \leq 1 \\ 1 & \text{se } v > 1 \end{cases}$$

e la stessa $f(v)$

$$f(v) = \begin{cases} 0 & \text{se } v < 0 \\ 1 & \text{se } 0 \leq v \leq 1 \\ 0 & \text{se } v > 1 \end{cases}$$

Utilizzando la proprietà vista a lezione, il valore atteso per ogni i sarà

$$E_{v_i \sim U[0,1]}[v_i] = \frac{i}{n+1}$$

quindi la sua valutazione virtuale sarà

$$\varphi_i(v_i) = \frac{i}{n+1} + \frac{i}{n+1} - 1 = \frac{2i}{n+1} - 1$$

Per questo modello d'aste sappiamo che massimizzare il profitto equivale a massimizzare il welfare. Dovrò quindi scegliere la mia allocazione come

$$\underset{x}{\text{argmax}}\ E_{v \sim U[0,1]}\Big[\sum_i \varphi_i \cdot x_i\Big]$$

Dovrò quindi guardare ad un prezzo di riserva $\frac{1}{2}$, che deriva da

$$\frac{2i}{n+1} - 1 \geq 0 \ \equiv \ \frac{i}{n+1} \geq \frac{1}{2}$$

Stabilisco quindi il prezzo usando Mayerson, quindi

$$p_i = \begin{cases} \max\{\frac{1}{2}, B\} & \text{se } x_i = 1 \\ 0 & \text{altrimenti} \end{cases}$$

dove $B = \max_j \varphi_j(v_j)$, come nell'esercizio 2.1.
L'unico valore che influisce è quindi il numero dei partecipanti.

4.3 Esercizio 3

Si consideri l'asta a singolo oggetto con n partecipanti in cui le valutazioni sono estratte secondo $F_1, \ldots, F_n$ *regolari. Si definiscano i pagamenti in funzione di* φ_i*, si fornisca un esempio in cui il partecipante con l'offerta più alta non vince e si fornisca una giustificazione a tale evento.*

La formula dei pagamenti è la stessa definita nell'esercizio precedente, l'unica differenza sta nelle F.
La dimostrazione è abbastanza intuitiva, abbiamo visto come nel caso generale $\varphi_i(v_i) = 2v_i - 1$ se $F = U[0, 1]$. Prendiamo ora $n = 2$ per semplicità, essendo il procedimento semplicemente algebrico è facile esetenderlo.
Supponiamo i_1 si presenti con $v_{i_1} = \frac{3}{4}$ mentre i_2 si presenti con $v_{i_2} = \frac{2}{3}$, calcoliamo le loro valutazioni virtuali

$$\varphi_{i_1}(v_{i_1}) = \frac{3}{2} - 1 = \frac{1}{2}$$

Poniamio ora $F_2 = U[\frac{1}{3}, \frac{2}{3}]$. Risolvendo come negli esercizi precedenti otteniamo $\varphi_i(v_i) = 2v_i^2 - \frac{1}{3}$, quindi

$$\varphi_{i_2}(v_{i_2}) = \frac{8}{9} - \frac{1}{3} = \frac{5}{9}$$

Anche se i_1 ha una valutazione più alta il partecipante che alloca è i_2.

5 Set 5 - Meccanismi senza scambio di denaro

5.1 Esercizio 1

Argomentare se **RANDOM PRIORITY REALLOCATION** *è DSIC e se garantisce, indipendentemente dell'ordine random scelto, l'assenza di blocking-coalitions.*

L'algoritmo **RPR** è DSIC perchè analogo a **TTC**, l'unica differenza è la rinomina dei partecipanti.
Supponiamo infatti che tutti siano stati sinceri, allora prendendo $i \in [n]$ avrò un'iterazione k in cui una casa viene assegnata ad i. L'unico modo in cui può migliorare, ovvero ottenere una casa che preferisce di più di quella ottenuta all'iterazione k, è che in una delle iterazioni precedenti a k il partecipante i facesse parte di un ciclo.
Dato che, se questa situazione fosse stata possibile, i sarebbe stato servito ad un'iterazione $j < k$, possiamo concludere che in nessuna iterazione $j < k$ il paretecipante i faceva parte di un ciclo. Quindi l'algoritmo alloca la casa migliore possibile per quelle che sono le disponibilità all'iterazione k, ad ogni partecipante conviene quindi essere sicero ed il meccanismo si dimostra DSIC.
L'algoritmo, però, non garantisce anche l'assenza di *blocking-coalition* in quanto non da lo stesso output di **TTC**. Sappiamo infatti che TTC è l'unico algoritmo di assegnamento che garantisce l'assenza di *blocking-coalitions* e che un algoritmo di assegnamento, per garantire la stessa proprietà, deve fornire lo stesso output di TTC.
Dato che eseguo una permutazione sui partecipanti modifico anche l'ordine in cui verranno serviti, rendendo quindi impossibile garentire lo stesso output di TTC per ogni possibile permutazione applicata.

5.2 Esercizio 2

Si presenti un'insieme di preferenze che induca l'algoritmo **DEFERRED ACCEPTANCE ALGORITHM** *ad eseguire in* $\Theta(n^2)$

L'insieme di preferenze in questione è quello in cui, per ogni ospedale, il primo tirocinante a presentarsi è quello per il quale l'ospedale ha la preferenza minore. Ogni tirocinante dovrà quindi scorrere tutti gli ospedali nella lista.

5.3 Esercizio 3

Si estenda il concetto di stable matching e **DEFERRED ACCEPTANCE ALGORITHM** *perchè permettano l'esistenza di un'opzione esterna.*

Dato un insieme n di tirocinanti $r_1, \ldots, r_n$ ed un insieme di n ospedali $h_1, \ldots, h_n$ dove ogni tircinante ha una lista di preferenze per gli ospedali, e viceversa.
Per estendere il concetto di *stable matching* dovremo estendere il concetto di *blocking pair*. Fissato un matching M, la coppia (r, h) è *blocking pair* per M se esiste la coppia (r', h') t.c. $(r, h'), (r', h) \in M$ e

$$(h \geq_r h' \wedge r \geq_h r') \vee (ext \geq_r h' \vee ext \geq_h r')$$

dove ext è l'opzione esterna, ovvero il restare *unmathced*.
L'algoritmo può essere modificato come segue

```
while ∃ r unmatched do
    r si offre ad h in ordine, se non lo ha già rifiutato o ext ≥_r h
    if h unamtched then
        if ext ≥_h r then
            h rifiuta r
        else
            M.add((h, r))
        end if
    else
        if r ≥_h r' then
            M.pop((h, r'))
            M.add((h, r))
        else
            h rifiuta r
        end if
    end if
end while
```

5.4 Esercizio 4

Si denoti con r_h^* il peggior tirocinante per l'ospedale h. Si dimostri che in **DEFERRED ACCEPTANCE ALGORITHM** *ogni h verrà sempre matcheato con il suo r_h^**

Supponiamo esista uno stable matching M in cui h viene matchato con un $r' \neq r_h^*$. Dato che, differenza dell'esercizio precedente, devo per forza accoppiare tutti i tirocinanti e tutti gli ospedali, allora $\exists\, h' : (r_h^*, h') \in M$.
Se $(r', h) \in M \wedge r_h^* \geq_h r'$ e $(r_h^*, h') \in M \wedge h \geq_{r_h^*} h'$ allora (r_h^*, h) è blocking coalition per M, di conseguenza M non può essere *stable matching*.

5.5 Esercizio 5

Si definiscano un profilo tirocinanti-ospedali sincero ed uno in cui un sopedale mente per migliorare la sua situazione

Definiamo il profilo dei tirocinanti, che resterà uguale per entrambi i profili degli ospedali

$$\begin{aligned} r_1 &\rightarrow h_3, h_2, h_1 \\ r_2 &\rightarrow h_1, h_3, h_2 \\ r_3 &\rightarrow h_3, h_1, h_2 \end{aligned}$$

e infine definiamo i due profili per gli ospedali

$$\text{Sincero}: \begin{aligned} h_1 &\rightarrow r_2, r_3, r_1 \\ h_2 &\rightarrow r_1, r_3, r_2 \\ h_3 &\rightarrow r_2, r_1, r_3 \end{aligned} \qquad \text{Non sicero}: \begin{aligned} h_1 &\rightarrow r_3, r_2, r_1 \\ h_2 &\rightarrow r_1, r_3, r_2 \\ h_3 &\rightarrow r_2, r_1, r_3 \end{aligned}$$

6 Set 6 - Selfish Routing, Makespan e No Regret

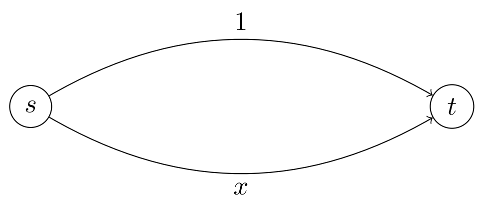
Vista l'eterogeneità degli argomenti trattati, ognuno avrà una propria sezione, in modo da organizzare al meglio gli esercizi.

6.1 Non-atomic Selfish Routing

6.1.1 Esercizio 1

Si dimostri che, data $C = \{c(x) = ax + b \mid a, b > 0\}$ il POA bound sulla Rete di Pigou è $\frac{4}{3}$

Data la *Rete di Pigou*



dove chiamiamo l'arco inferiore l e quello superiore u.
Dato un *flusso di equilibrio* f, lo faccio passare interamente per l'arco inferiore, quindi $f_l = 1$. Da questo $C(f) = 1$.
Definisco il *flusso ottimo* f^* ed il suo costo

$$C(f^*) = \underset{x}{\text{argmax}}\ x^2 - x + 1 \ \Rightarrow\ \frac{1}{4} - \frac{1}{2} + 1 = \frac{3}{4}$$

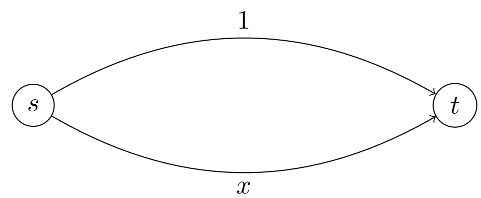
Per definizione il *Price of Anarchy* (POA) è il rapporto tra flusso di equlibrio e flusso ottimo, da questo

$$POA = \frac{C(f)}{C(f^*)} = \frac{1}{\frac{3}{4}} = \frac{4}{3}$$

6.1.2 Esercizio 2

Si dimostri che, in ogni Selfish Routing Network con funzioni di costo affini $c(x) = ax + b$, il flusso ottimo f^* smista il traffico in cammini il cui costo è al massimo il doppio di un cammino di costo minimo.

Sappiamo, per il teorema visto a lezione, che qualsiasi ricerca di bound su problemi di *Selfish Routing* può essere ridotto alla ricerca su reti *simil-Pigou*.
Nello specifico, per la classe $\mathbb{C} = \{c(x) \mid c(x) = ax+b,\ a, b > 0\}$, il caso peggiore è proprio quello di Pigou



Nell'esercizio precedente abbiamo visto come un flusso ottimo smisti equamente in traffico tra l e u, infatti avendo traffico x su un nodo il nodo restante si troverà traffico $(1 - x)$.
Risolvendo come prima otterremo

$$x_l \cdot c(x_l) + (1 - x_l) \cdot c(1 - x_l) = x_l^2 - x_l + 1$$

Derivando otteniamo

$$x_l = \frac{1}{2}$$

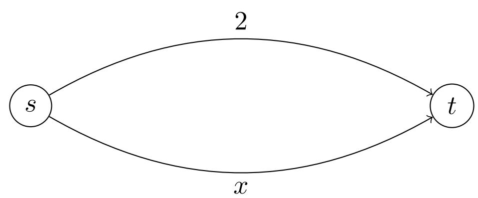
Quindi f^* equidistribuisce il traffico sui due archi.
Il flusso di equilibrio f ridireziona tutto su un singolo arco, portando il traffico ad 1. Questo raddoppia il traffico rispetto all'ottimo, dimostrando quanto detto nella consegna.

6.2 Atomic Selfish Routing

6.2.1 Esercizio 3

Si dia un upper bound per una rete di Atomic Selfish routing con 2 agenti e funzioni di costo affini che sia inferiore a $\frac{5}{2}$

Prendiamo come esempio la *Rete di Pigou* che segue



Ricordiamo che, a differenza della versione non atomica, ogni agente qui controlla un'unità di traffico, quindi se entrambi utilizzassero u il costo sarebbe $1 + 1 = 2$.
È possibile individuare due equilibri

1. I due agenti vanno su archi diversi, quindi i_u paga 2 e i_l paga 1, costo totale 3.

2. Entrambi utlizzano l'arco l, ciascuno paga 2 ed il costo totale sarà $2+2 = 4$.

Dato che il *Price of Anarchy* è definito come

$$\frac{\text{Costo del peggior equilibrio}}{\text{Costo dell'ottimo}}$$

non esistendo altre possibilità oltre alle due descritte, *1.* è il nostro ottimo mentre *2.* è il peggior equlibrio. Quindi $POA = \frac{4}{3}$

6.2.2 Esercizio 4

Data la funzione potenziale

$$\Phi(f) = \sum_{e \in E} \left(f_e \cdot c_e(f_e) + \sum_{i \in S_e} w_i \cdot c_e(w_i) \right)$$

dove S_e è l'insieme degli agenti che utilizzano l'arco e e w_i è il peso di ogni agente (fino ad ora abbiamo considerato $w_i = 1$), si dimostri che esiste almeno un PNE per ogni Atomic Selfish Routing Network.

Applichiamo delle trasformazioni a $\Phi(f)$ per portarla in una notazione più comoda ai fini della dimostrazione

$$\begin{aligned}
\Phi(f) &= \sum_{e \in E} \left(f_e \cdot c_e(f_e) + \sum_{i \in S_e} w_i \cdot c_e(w_i) \right) \\
&= \sum_{i=1}^{k} w_i \cdot \sum_{e \in P_i} c_e(f_e) + \sum_{e \in E} \sum_{i \in S_e} w_i \cdot c_e(w_i)
\end{aligned}$$

Rifacciamoci ora alla definizione di PNE e definiamo con $\hat{f}$ il flusso su cui devio. Definiamo la differenza nella funzione potenziale Φ indotta da questa deviazione come

$$\begin{aligned}
\Phi(\hat{f}) - \Phi(f) &= \sum_{i=1}^{k} w_i \sum_{e \in \hat{P}_i} c_e(\hat{f}_e) + \sum_{e \in E} \sum_{i \in S_e} w_i c_e(w_i) - \sum_{i=1}^{k} w_i \sum_{e \in P_i} c_e(f_e) + \sum_{e \in E} \sum_{i \in S_e} w_i c_e(w_i) \\
&= \sum_{i=1}^{k} w_i \cdot \sum_{e \in \hat{P}_i} c_e(\hat{f}_e) - \sum_{i=1}^{k} w_i \cdot \sum_{e \in P_i} c_e(f_e) \\
&= r \cdot \left(\sum_{e \in \hat{P}_i} c_e(\hat{f}_e) - \sum_{e \in P_i} c_e(f_e) \right)
\end{aligned}$$

È possibile vedere come, mantenendo per comodità le trasformazioni appena applicate, che la parte sinistra corrisponde all'aumento di un agente sugli archi del cammino, quindi

$$r \cdot \left(\sum_{e \in \hat{P}_i} c_e(f_e + 1) - \sum_{e \in P_i} c_e(f_e) \right)$$

Da questo, prendendo f tale che

$$f = \underset{f}{\text{argmin}} \ \Phi(f)$$

si può vedere che nessuna variazione di strategia unilaterale può ridurre ulteriormente Φ. Quindi f è PNE.