- **AVVERTENZA**: Di seguito trovate alcuni appunti, poco ordinati e poco formali, che uso come traccia durante le lezioni. **Non sono assolutamente da considerarsi sostitutivi del materiale didattico.**

- **Riferimenti**: Paragrafi 10.3 e 10.4 di Algorithmic Game Theory (pagine 253-258).

- Un gioco cooperativo, *con utilità non trasferibile*, è definito da una coppia (N, v). N è l'insieme dei giocatori; V è una funzione che associa ad ogni coalizione S di N un insieme $V(S) \subset \mathcal{R}_+^S$ di allocazioni ammissibili (vettori di payoff) per S. Si noti che anche per la grande coalizione abbiamo un insieme dato di possibili allocazioni ammissibili $V(N)$: vogliamo capire se qualcuna di queste allocazioni è stabile rispetto tutte le coalizioni. Questo ci porta al concetto di nucleo in questo caso:

- Il *nucleo* di un gioco cooperativo con utilità non trasferibile è l'insieme dei allocazioni $\alpha \in V(N)$ che soddisfano la seguente proprietà :

  – non esistono una coalizione $S \neq \emptyset$ e un vettore di utilità $x \in V(S)$ tale che: 1) $x_i \geq \alpha_i$ per ogni $i \in S$; 2) $x_j > \alpha_j$ per almeno un $j \in S$.

- È facile vedere che un gioco cooperativo (N, v) con utilità trasferibile può essere visto come un particolare gioco cooperativo con utilità non trasferibile (N, V) ponendo $V(S) = \{x \in \mathcal{R}_+^S : \sum_{i \in S} x_i \leq v(S)\}$ e $V(N) = \{x \in \mathcal{R}_+^N : \sum_{i \in N} x_i = v(N)\}$.

  In particolare, le due definizione di nucleo sono consistenti. Innanzitutto consideriamo un vettore α che non sia nel nucleo di (N, v): mostreremo che non è neanche nel nucleo del gioco (N, V). Naturalmente, se $\sum_{i \in N} \alpha_i \neq v(N)$, allora α non neanche è nel nucleo del gioco (N, V), poiché $\alpha \notin V(N)$. Supponiamo quindi che $\sum_{i \in N} \alpha_i = v(N)$, segue quindi che $\alpha \in V(N)$. Ora, se α non è nel nucleo di (N, v), allora esiste una coalizione S tale che $v(S) > \sum_{i \in S} \alpha_i$. Ma allora esiste un $\varepsilon > 0$ opportuno tale che il vettore $x \in \mathcal{R}_+^S$, con $x_i = \alpha_i + \varepsilon, i \in S$, soddisfa $v(S) \geq \sum_{i \in S} x_i$ e quindi $x \in V(S)$. Ma allora, poiché $x_i > \alpha_i$ per ogni $i \in S$, e $x \in V(S)$, segue che α non neanche è nel nucleo del gioco (N, V).

  Viceversa, consideriamo un vettore α che non sia nel nucleo di (N, V). Se $\alpha \notin V(N)$, allora $\sum_{i \in N} \alpha_i \neq v(N)$ e quindi α non neanche è nel nucleo del gioco (N, v). Supponiamo quindi che $\alpha \in V(N)$, cioè $\sum_{i \in N} \alpha_i = v(N)$. Allora, se α non è nel nucleo di (N, V), esistono una coalizione $S \neq \emptyset$ e un vettore di utilità $x \in V(S)$ tali che 1) $x_i \geq \alpha_i$ per ogni $i \in S$; 2) $x_j > \alpha_j$ per almeno un $j \in S$. Quindi $\sum_{i \in S} \alpha_i < \sum_{i \in S} x_i$, e poiché $x \in V(S)$, $\sum_{i \in S} x_i \leq v(S)$. Segue che $\sum_{i \in S} \alpha_i < v(S)$, ovvero α non è neanche nel nucleo di (N, v).

1 The house allocation problem

- Studiamo adesso un problema che rientra nell'alveo dei giochi cooperativi con utilità non trasferibile: *The house allocation problem* (HAP).

  In questo gioco è dato un insieme N di n giocatori e un insieme C di n case. Ogni giocatore possiede una e una sola delle case; inoltre ogni giocatore ha in mente una graudatoria di tutte le n case dell'insieme C: questa graudatoria è un ordine totale: ovvero, ogni giocatore i, per ogni coppia di case $u, v \in C$, preferisce u a v oppure v a u. L'obiettivo è quello di riallocare le case tra i giocatori, ovvero trovare un matching M di dimensione n, che sia *stabile*. Al solito, la stabilità è per noi rispetto la possibilità che una coalizione S, cioè un sottoinsieme di giocatori, rompano il matching M trovando "all'interno" di S una soluzione che per i giocatori di S sia preferibile a quella offerta da M.

  Per ogni insieme S, di giocatori definiamo dunque $V(S)$ come un qualunque *matching* che assegna a ciascun giocatore di S una e una sola casa dell'insieme $C(S) := \{c \in C : c \text{ appartiene a un giocatore } i \in S\}$. Per esempio, $V(N)$ è cosituito da tutti i matching di dimensione n tra giocatori e case.

  A questo punto vorremmo individuare un matching $M \in V(N)$, che assegna quindi a ciascun giocatore di N una e una sola casa dell'insieme C, in modo tale che sia soddisfatta la seguente proprietà (P):

  non esistono una coalizione $S \neq \emptyset$ e un matching $M' \in V(S)$, M' assegna quindi a ciascun giocatore di S una e una sola casa dell'insieme $C(S)$, tali che:

  - per ogni giocatore $i \in S$, il matching M' assegna a i una casa non peggiore (secondo le preferenze di i) di quella che gli assegna il matching M;
  - per almeno un giocatore $j \in S$ il matching M' assegna a j una casa migliore di quella che gli assegna il matching M

  ovvero la stabilità al solito è intesa rispetto alla tentazione di una coalizione $S \subset N$ di rompere la grande coalizione.

  Naturalmente, è immediato verificare come questo problema rientri sostanzialmente nel formato dei giochi cooperativi con utilità non trasferibile. La differenza principale risiede nel fatto che l'insieme $v(S)$ è definito come un insieme di oggetti combinatori, matching in particolare, piuttosto che come un insieme di punti dello spazio $\mathcal{R}^S_+$.

  Chiameremo quindi *nucleo* del gioco HAP l'insieme di tutti i matching di dimensione n che soddisfano la proprietà (P).

- Esempio. 8 Giocatori $\{1, 2, 3, 4, 5, 6, 7, 8\}$. Supponiamo ogni giocatore i possiede la casa i. Ecco le graudatorie di ciascun giocatore:

  - Giocatore 1: $\{3, \ldots\}$;

- Giocatore 2: $\{5, 4, 7, \ldots\}$;
- Giocatore 3: $\{5, \ldots\}$;
- Giocatore 4: $\{6, 1, 5, 8, \ldots\}$;
- Giocatore 5: $\{1, \ldots, \}$;
- Giocatore 6: $\{6, \ldots\}$;
- Giocatore 7: $\{8, 3, 5, 7, \ldots\}$;
- Giocatore 8: $\{4, 5, 1, \ldots\}$;

- Come vediamo nel seguito, il nucleo del gioco HAP è costituitoda uno e un solo matching. Questo matching è individuato da un algoritmo che va sotto il nome di *Top Trading Cycle Algorithm* (TTCA) .

  Costruiamo un grafo orientato G come segue. I nodi di G corrispondono ai giocatori di N e ogni nodo ha esattamente un successore: corrispondente al proprietario della casa che i preferisce tra quelle dell'insieme N. Si noti che l'unico arco uscente da i est l'arco (i, i), se la casa preferita da i è quella che possiede: in questo caso parliamo di *loop*. Nel seguito i loop di G saranno considerati a tutti gli effetti dei particolari *cicli* di G. Il grafo G per costruzione ha n nodi e n archi. Tralasciamo per un attimo gli orientamenti e guardiamo il grafo non orientato $\bar{G}$ "sotteso" da G. Un grafo non orientato con n nodi che non ha cicli ha al più $n - 1$ archi, quindi esiste almeno un ciclo in $\bar{G}$. Ora osserviamo che, se consideriamo nuovamente G, ogni suo ciclo C è orientato (ovvero possiamo percorrere tutto il ciclo rispettando gli orientamenti degli archi), altrimenti ci dovrebbe essere almeno un nodo di C che ha due archi (del ciclo) uscenti, il che non è possibile per costruzione di G (ogni nodo ha esattamente un arco uscente). Possiamo quindi concludere che l'insieme dei cicli di G è non vuoto, è formato da cicli orientati, e questi non hanno nodi in comune (di nuovo per giustificare quest'ultima osservazione è sufficiente ricordare che ogni nodo di G ha esattamente un arco uscente). Allora possiamo individuare un primo matching parziale, indotto dai cicli di G: sia N_1 l'insieme dei nodi appartenenti ad un ciclo di G, assegniamo a ogni giocatore $i \in N_1$ la casa posseduta giocatore $j \in N_1$ tale che $(i, j) \in E(G)$. Rimuoviamo quindi dal gioco i giocatori dell'insieme N_1 e le case dell'insieme $C(N_1)$.

  A questo punto costruiamo un nuovo grafo G_1 come segue. L'insieme dei giocatori (nodi) è $N \setminus N_1$, e nuovamente ogni nodo i ha un unico successore, corrispondente al proprietario della casa che i preferisce *tra quelle dell'insieme* $N \setminus N_1$. Ragionando come prima, l'insieme dei cicli di G_1 è non vuoto, è formato da cicli orientati, e questi non hanno nodi in comune. Allora possiamo individuare un secondo matching parziale, indotto dai cicli di G_1: sia N_2 l'insieme dei nodi appartenenti ad un ciclo di G_1, assegniamo a ogni giocatore $i \in N_2$ la casa posseduta giocatore $j \in N_2$ tale che $(i, j) \in E(G_1)$. Rimuoviamo quindi dal gioco i giocatori dell'insieme N_2 e le case dell'insieme $C(N_2)$ ... in al più $k \leq n$ iterazioni l'algoritmo termina restituendoci un matching M^* di dimensione n.

- Osserviamo che nel caso precedente le case assegnate ai giocatori $1, 2, 3, 4, 5, 6, 7, 8$ dal TTCA sono rispettivamente: 3,2,5,8,1,6,7,4.

- Come mostriamo nel seguito, l'algoritmo TTCA resituisce sempre è l'unico matching nel nucleo del gioco.

  Innanzitutto osserviamo che non ci possono essere altre soluzioni del nucleo diverse da quella restituita da TTCA. Consideriamo una soluzione M che sia nel nucleo. Osserviamo che perché M sia stabile rispetto la coalizione N_1, M *deve* assegnare i giocatori di N_1 esattamente come TTCA: quindi M assegna i giocatori di N_1 come TTCA. Passiamo ai giocatori di N_2. Poiché le case dei giocatori di N_1 sono già assegnate, M assegna i giocatori di N_2 a case possedute da giocatori di $N \setminus N_1$, allora perché M sia stabile rispetto la coalizione $N_1 \cup N_2$, M *deve* assegnare i giocatori di N_2 esattamente come TTCA $\ldots$ Quindi M deve assegnare le case come TTCA e non ci possono essere nel nucleo matching diversi dal matching M^* prodotto da TTCA.

  Mostriamo ora che il matching M^* prodotto da TTCA è effettivamente nel nucleo. Consideriamo una coalizione S e partizioniamola nelle classi $S \cap N_1, S \cap N_2, \ldots, S \cap N_k$ (dove k è il numero di iterazioni svolte dall'algoritmo TTCA). Se M^* non fosse stabile, esisterebbe un matching M_S (tra i giocatori di S e le case di $C(S)$) e un giocatore $i \in S$ tale che M_s assegna a i una casa migliore di quella assegnatagli da M^*, e assegna a ogni giocatore di $S \setminus i$ una casa non peggiore di quella assegnatagli da M^*. Consideriamo i giocatori di $N_1 \cap S$, essi ottengono da M^* la casa preferita, quindi devono continuare a ottenere da M_s la casa preferita (ovviamente non possono migliorare), quindi M_s e M^* definiscono lo stesso, matching per i giocatori di $N_1 \cap S$. Ma allora possiamo ripetere il ragionamento svolto in precedenza per mostrare che l'algoritmo TTCA restituisce sempre è l'unico matching nel nucleo del gioco, e concludere che su ogni livello i M_s e M^* devono definire lo stesso matching per i giocatori di $N_i \cap S$, il che e' una contraddizione.

- L'algoritmo TTCA è un *meccanismo* che assegna a ciascun giocatore un payoff (una casa dell'insieme C) sulla base delle preferenze (le graudatorie) indicate da ciascun giocatore. Questo ci ricorda un po' i meccanismi d'asta. In quel caso ci ponevamo l'obiettivo di disegnare un meccanismo che fosse *truthful revealing* (oppure come si suol dire, *strategy-proof*), ovvero che portasse ogni giocatore a porre nella busta il vero valore che egli/ella attribuiva al bene oggetto dell'asta. In questo caso, possiamo chiederci: il meccanismo TTCA induce ogni giocatore a riportare la sua vera graduatoria? O viceversa il meccanismo porterebbe vantaggio a un giocatore che decidesse di comunicare una graduatoria falsa? Osserviamo come questa questione prescinda dal fatto che M^* sia nel nucleo del gioco. Posso ignorarlo e comunque chiedermi: se decido di riassegnare le case secondo TTCA, queso meccanismo è strategy-proof?

  La risposta è affermativa. Consideriamo il caso in cui il giocatore i riporta la sua vera graduatoria e gli altri giocatori riportano le loro graduatorie (vere o false, non importa). Il meccanismo assegna una casa h al giocatore i; ci chiediamo, ferme restando le graduatorie degli altri giocatori, a i potrebbe convenire cambiare la sua graduatoria? Per rispondere a questa domanda supponiamo che TTCA abbia assegnato una casa ad i alla iterazione j, con $1 \leq j \leq k$. Notiamo che a ciascun giocatore h dei livelli $N_1, \ldots < N_{j-1}$ è stata assegnata una casa che il giocatore h preferisce a quella posseduta da i, e il fatto che i cambi la *sua* graduatoria è per h indifferente. Quindi anche cambiando la sua graduatoria ad i non sarebbe assegnata una casa tra quelle possedute dai giocatori in $N_1 \cup \ldots \cup N_{j-1}$. D'altro canto aver dichiarato la vera graduatoria gli garantisce che gli venga assegnata la casa che lui preferisce tra quelle possedute dai giocatori in $N \setminus (N_1 \cup \ldots \cup N_{j-1})$, quindi cambiare la graduatoria non migliorerebbe la casa che gli viene assegnata.

2 The stable marriage problem

- In questo problema sono dati $2n$ giocatori: n uomini e n donne. Ogni giocatore ha in mente una graudatoria di tutte le n persone dell'altro sesso: questa graudatoria è un ordine totale. L'obiettivo è quello di accoppiare gli uomini con le n donne, ovvero trovare un matching M di dimensione n, che sia *stabile*. Al solito, la stabilità è per noi rispetto la possibilità che una coalizione S, cioè un sottoinsieme di giocatori, rompano il matching M trovando all'interno di S una soluzione che per i giocatori di S sia preferibile a quella offerta da M.

  Nella descrizione del gioco stiamo implicitamente assumendo che ogni giocatore preferisca in ogni caso accoppiarsi che rimanere single: è possibile estendere facilmente il gioco in modo da considerare questa possibilità (così come la possibilità che il numero di uomini e donne sia diverso), ma non ce ne occupiamo. Con la nostra assunzione, naturalmente, per verificare che il matching M sia stabile, ha senso considerare solo coalizioni S con lo stesso numero di giocatori uomini e donne.

- Diciamo che un matching M di dimensione n è quindi stabile se non esiste un insieme S di $h \leq n$ uomini e h donne e un matching M' tra gli uomini e le donne di S tale che:

  - per ogni giocatore $i \in S$, il matching M' assegna a i un partner non peggiore (secondo le preferenze di i) di quella che gli assegna il matching M;
  - per almeno un giocatore $\bar{i} \in S$ il matching M' assegna a $\bar{i}$ un partner migliore di quella che gli assegna il matching M.

  Osserviamo ora che se esiste un matching M' come sopra, allora esistono necessariamente un uomo m e una donna w tali che, detti rispettivamente $w^M(m)$ la donna assegnata dal matching M a m e $m^M(w)$ l'uomo assegnato a w dal matching M, m preferisce w a $w^M(m)$, e w preferisce m a $m^M(w)$. Naturalmente, vale anche il viceversa; ovvero, se esistono un uomo m e una donna w tali che m preferisce w al partner $w^M(m)$ assegnatogli da M, e w preferisce m al partner $m^M(w)$ assegnatole da M, allora il matcing M non è stabile rispetto la coalizione $S = \{m, w\}$. Possiamo quindi concludere che:

Un matching M di dimensione n è stabile se e solo se non esistono un uomo m e una donna w tali che m preferisce w al partner $w^M(m)$ che il matching M gli assegna, e w preferisce m al partner $m^M(w)$ che il matching M le assegna.

E, come al solito, il nucleo di questo gioco è costituito da tutti i matching stabili.

- Come dimostriamo nel seguito, esiste sempre un matching stabile, ovvero il nucleo di questo gioco è sempre non vuoto.

- L'algoritmo di Gale-Shapley.

  Alla prima iterazione ogni uomo si propone alla donna che preferisce. Ogni donna considera tutte le proposte che ha ricevuto e dice: "Forse" all'uomo che preferisce tra quelli che le si sono proposti (se ce ne sono); "No" a tutti gli altri uomini che le si sono proposti. Le donne che hanno ricevuto un'offerta sono temporaneamente fidanzate (all'uomo a cui hanno detto "Forse").

  In ogni iterazione successiva, ogni uomo non fidanzato si propone alla donna che preferisce, tra quelle che non lo hanno già rifiutato (la donna può essere o non essere già fidanzata. Di nuovo la donna esamina tutte le proposte fin qui ricevute (e non già rifiutate) e risponde "Forse" all'uomo che preferisce e "No" agli altri (includendo in questa analisi il suo eventuale fidanzato corrente). Questo vuol dire che una donna può cambiare fidanzato, e un uomo può essere mollato.

  Iteriamo questo fino a quando tutte le donne (e quindi anche tutti gli uomini) sono temporaneamente fidanzate. Questi fidanzamenti sono il matching finale M^* proposto dall'algoritmo.

  Esempio. 4 uomini: $\{1, 2, 3, 4\}$ e 4 donne: $\{1, 2, 3, 4\}$.

  Preferenze uomini. 1: 2,1,3,4. 2: 4,1,2,3. 3: 1,3,2,4. 4: 2,3,1,4.

  Preferenze donne. 1: 1,3,2,4. 2: 3,4,1,2. 3: 4,2,3,1. 4: 3,2,1,4.

  (per lo svolgimento dell'algoritmo si veda [**?**])

- L'algoritmo termina in al più n^2 iterazioni. A ogni iterazione, tranne quella finale, almeno un uomo viene rifiutato. Quest'uomo alla fine cancella la donna che lo ha rifiutato dalla propria graduatoria. Poiché ci sono n uomini e ognuno ha una graduatoria con n donne Banalmente, in al più n^2 iterazioni l'algoritmo termina.

  Il matching è stabile. Facciamo un'osservazione preliminare: se una donna è temporaneamente fidanzata a una certa iterazione, lo sarà anche nell'iterazione successiva, e con un uomo non peggiore (secondo la sua graduatoria). Siano ora w una donna e m un uomo non accoppiati da M. Al termine dell'algoritmo non è possibile che entrambi preferiscano l'altro/a al partner asegnato dall'algoritmo. Se m preferisce w al partner $w^{M^*}(m)$, egli si deve essere proposto a w in qualche iterazione dell'algoritmo. Se in quella iterazione w aveva accettato (temporaneamente) la proposta di m, poiché alla fine non è fidanzata con m, vuol dire che lo ha mollato in qualche iterazione sucessiva per qualcuno che ella preferisce, e, per via dell'osservazione precedente, anche alla fine sarà fidanzata a qualcuno che ella preferisce. Viceversa, se in quella iterazione w non aveva accettato (temporaneamente) la proposta di m, era già fidanzata a qualcuno che ella preferisce, e di nuovo per via dell'osservazione precedente, anche alla fine sarà fidanzata a qualcuno che ella preferisce.

- Osserviamo che, per l'esempio precedente, se avessimo svolto l'algoritmo di Gale-Shapley a partire dalle donne avremmo identificato un *diverso* matching stabile (provare!). In generale, esistono più matching stabili e in particolare ci possono essere matching stabili che sono diversi da quello prodotto dall'algoritmo di Gale-Shapley partendo dagli uomini e da quello prodotto dallo stesso algoritmo partendo dalle donne.

- Anche l'algoritmo di Gale-Shapley è un *meccanismo* che assegna a ciascun giocatore un payoff (un partner) sulla base delle preferenze (le graudatorie) indicate da ciascun giocatore. Come vediamo nel seguito, questo meccanismo (algoritmo), se svolto a partire dagli uomini, non è sempre *truthful revealing* per le donne, ovvero una donna potrebbe avere convenienza a riportare una graduatoria falsa; mentre, si può dimostrare che il meccanismo è truthful revealing per gli uomini. Analogamente se questo meccanismo (algoritmo) è svolto a partire dalle donne, è per queste un meccanismo truthful revealing, mentre non vale lo stesso per gli uomini.

  Esempio. 3 uomini: $\{A, B, C\}$ e 3 donne: $\{X, Y, Z\}$.

  (Vere) Preferenze uomini. A: X, Y, Z. B: X, Z, Y. C: Y, X, Z.

  (Vere) Preferenze donne. X: C, A, B. Y: A, C, B Z: A, B, C.

  L'algoritmo di Gale Shapley, svolto a partire dagli uomini, restituisce il matching: AX, BZ, CY.

  Ma se la donna X avesse mentito e riportato C, B, A, (e tutti gli altri invece si fossero comportati onestamente, l'algoritmo di Gale Shapley, svolto a partire dagli uomini, riporta un altro matching stabile che : AY, BZ, CX, che per X è meglio!

  È facile invece convincersi che l'algoritmo, svolto a partire dagli uomini, è per questi truthful revealing.

References

[1] http://en.wikipedia.org/wiki/Stable_marriage_problem