Introduzione teorica ai modelli Operations Research Game

Stefano Moretti

Dipartimento di Matematica, Universita' di Genova

Email: moretti@dima.unige.it

Phone:010-3536838

RO e TdG

- Forti connessioni tra Ricerca Operativa (RO) e la Teoria dei Giochi (TdG) (es. dualità nella programmazione matematica e minimax per i giochia a somma zero)
- negli ultimi 30 anni l'interazione tra le due discipline sembra essersi concentrata tra la RO e la TdG cooperativi.

Ricerca Operativa

- Un decisore, guidato da una funzione obiettivo, affronta un problema di ottimizzazione.
- La teoria quindi si concentra sulla questione di come agire in maniera ottimale e, in particolare, sulla complessità computazionale ed il design di algoritmi efficienti.

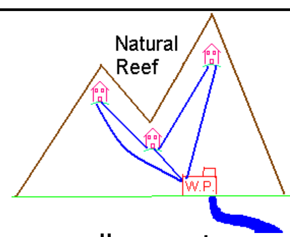
Teoria dei Giochi cooperativi

- almeno due decisori interagenti (chiamati **giocatori**)
- sono permessi **accordi vincolanti**
- possono essere permessi anche **pagamenti laterali** (giochi a utilità trasferibile o **TU-game**)

RO e TdG→ORG

- Struttura (discreta) di base di un grafo, network o sistema che soggiace a varie tipologie di **problemi di ottimizzazione** combinatoria.
- Si assume che almeno due **giocatori** sono situati **in corrispondenza di parti** (es. vertici, lati, panieri di risorse, lavori) del sistema da ottimizzare ecc.)

Esempio di **Situazione**



Si consideri:

- Un gruppo di persone le cui case sulla montagna non siano ancora connesse ad una rete fognaria;
- Le loro acque reflue devono essere raccolte in un depuratore a valle;
- Per tutti e' sufficiente, ma non necessario, essere connessi autonomamente al depuratore;
- Ci si puo connettere anche attraverso altre case;
- "Alcune connessioni potrebbero anche essere impedite da barriere naturali (natural reef)";
- Costruire un tubo e' costoso.

Come nasce il gioco?

- **Lavorando assieme**, i giocatori possono eventualmente realizzare **guadagni extra** o **abbassare i costi** in comparazione alla situazione in cui ciascuno ottimizza individualmente.
- La questione nuova che emerge è **come dividere**, ripartite i guadagni extra o i risparmi.

Ricordo che

Un gioco coperativo dei costi e' una coppia ordinata <N,c> dove

$N=\{1,2,\ldots,n\}$ e' l'insieme dei giocatori

$c:2^N \rightarrow \Re^+$ e' la funzione caratteristica del gioco

che assegna ad ogni coalizione $S \in 2^N$ un numeor reale $c(S)$ e dove $c(\{\emptyset\})=0$.

Un vettore $x \in \Re^n$ e' chiamato *allocazione*

Se un'allocazione e' sia *efficiente* ($\Sigma_{i \in N} x_i = c(N)$) che *individualmente razionale* ($x_i \leq c(\{i\})$ per ogni $i \in N$) allora e' chiamata *imputazione*

Se un'imputazione e' anche *stabile* ($\Sigma_{i \in S} x_i \leq c(S)$ for each $S \subseteq N$) allora e' chiamata *allocazione del nucleo*.

Il *nucleo* di un gioco <N,c> e' l'insieme di tutte le allocazioni del nucleo ed e' denotato da Core(N,c)

Proprieta' "banali" di giochi cooperativi dei costi

Un gioco cooperativo dei costi si dice subadditivo se

$$c(S)+c(T) \geq c(S \cup T)$$

per ogni $S,T \in 2^N$ con $S \cap T = \varnothing$ (e' detto fortemente subadditivo se vale il maggiore stretto nella disuguaglianza).

Si noti il segno invertito rispetto alla superadditivita' dei giochi di guadagno.

Si noti inoltre che un gioco dei costi <N,c> puo' essere visto come gioco di guadagni in almeno due modi diversi:

- si considera il gioco <N,v> in cui $v(S)=-c(S)$ per ogni $S \in 2^N$;
- si considera il gioco dei risparmi $\langle N,v_R \rangle$ in cui

$$v_R(S)= (\Sigma_{i \in S}\, c(\{i\}))-c(S) \text{ per ogni } S \in 2^N.$$

ORG: classe molto molto variegata

- Basati su problemi di ottimizzazione su grafi:
  - **problemi di connessione**, cioè situazioni che coinvolgono reti di connessione in situazioni di cooperazione.
  - **Problemi *routing***, ovvero problemi nei quali l'obiettivo è quello di trovare un cammino di minimo costo su un grafo pesato.
- Basati sul concetto economico di mercato:
  - **Problemi di *scheduling***, in cui la funzione caratteristica dipende dalla posizione dei giocatori in una coda.
  - **Problemi di produzione**, basati sulla potenzialità di produzione dei giocatori.
- modelli di RO stocastici in cui la funzione obiettivo da ottimizzare è ex-ante affetta da incertezza

Problemi di connessione

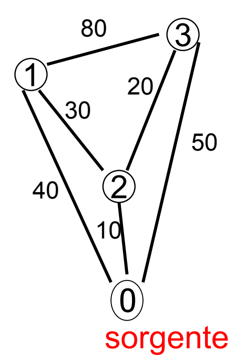
- fixed tree games, ovvero giochi derivanti da problemi di mantenimento di network già costruiti
- minimum cost spanning tree games (mcst games), dove invece il network di connessione deve ancora essere realizzato.

Minimum Cost Spanning Tree Situation

Utilizziamo il modello del grafo pesato completo.

- I cui vertici rappresentano le case
- il vertice 0 e' la sorgente
- I lati rappresentano le connessioni

- I numeri vicino ai lati rappresentano il costo di connessione



Minimum Cost Spanning Tree Situation: $<N',E_{N'},w>$

$N=\{1,2,\dots,n\}$ insieme degli agenti

$N'= N\cup\{0\}$, dove 0 e' la sorgente

$E_{N'}= \{ \{i,j\} \mid i,j\in N' \text{ and } i\neq j\}$ e' l'insieme degli archi o lati

$<N',E_{N'}>$ e' un grafo completo con

$N\cup\{0\}$ come insieme dei nodi

$E_{N'}$ come insieme dei lati

$w: E_{N'}\rightarrow\Re^+$ e' una funzione non negativa sull'insieme dei lati

Minimum Cost Spanning Tree problem.

Problema di Ottimizzazione:

come connettere ogni nodo alla sorgente 0 in maniera tale che il costo di costruzione di del network di ricoprimento (che connette tutti i nodi direttamente o indirettamente alla sorgente 0) sia minimo?

Esempio

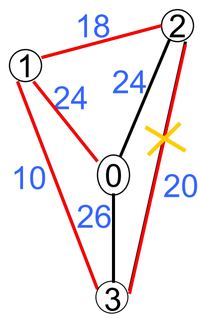
Sia $<N',E_{N'},w>$ un MCST situation tale che

$N=\{1,2,3\}$

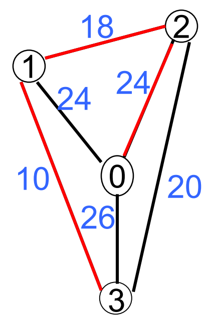
$E_{N'}=\{(1,0),(2,0),(2,1),(3,0),(3,1),(3,2)\}$

Una funzione dei costi come indicata sul grafo

Algoritmo di Kruskal



Algoritmo di Prim



$\Gamma_N=<N',\{\{1,0\},\{3,1\},(1,2)\}>$
e
$\Gamma^{\#}_N=<N',\{\{2,0\},\{3,1\},(1,2)\}>$
sono MCST per $<N',E_{N'},w>$;
il loro costo e’ pari a
10+18+24=52

Algoritmo di Prim.

INPUT: $<N',E_{N'},w>$ MCST situation.
OUTPUT: un MCST per $<N',E_{N'},w>$

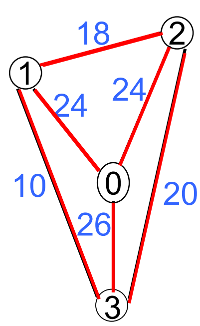
- *Step 1*: formare un lato di minimo costo da 0 a un vertice in N; tale vertice risulta cosi’ connesso a 0.
- FOR j={2,…,|N|} DO
- BEGIN
  - *Step j*: formare un lato di minimo costo tra un vertice in N non ancora connesso alla sorgente 0, direttamente o indirettamente, e la sorgente 0 o un vertice gia’ connesso ad essa, direttamente o indirettamente, negli step *i* precedenti, $i \in \{1,\dots j-1\}$
- END

Algoritmo di Kruskal.

INPUT: <N',E$_{N'}$,w> MCST situation.
OUTPUT: un MCST per <N',E$_{N'}$,w>

- *Step 1*: formare un lato di minimo costo tra i vertici in N∪{0}.
- FOR j={2,…,|N|} DO
- BEGIN
  - *Step j*: formare un lato di minimo costo tra i vertici in N∪{0} che non formino cicli con i lati costruiti negli step *i* precedenti, *i*∈{1,…*j*-1}
- END

Esempio: Il gioco cooperativo dei costi <{1,2,3},c> dato dalla mcst situation precedente e' tale che:



c(1)=24

c(2)=24

c(3)=26

c(1,3)=34

c(1,2)=42

c(2,3)=44

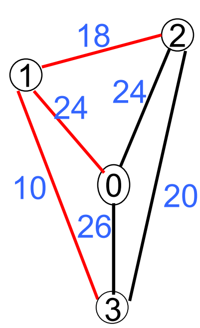
c(1,2,3)=52

Prop. Un mcst game <N,c> e' fortemente subadditivo.

Allocazione di Bird

- Costruito un mcst Γ si definisce l'immediato predecessore su Γ di un giocatore $i \in N$ un altro giocatore $j \in N \setminus \{i\}$ tale che
  - j si trova sul percorso da i alla sorgente 0;
  - $\{i,j\}$ e' un lato di Γ.
- L'allocazione di Bird rispetto a Γ assegna a ciascun giocatore i il costo $w(\{i,j\})$ del lato $\{i,j\}$.

Example: The cooperative cost game game <{1,2,3},c> on the MCST situation is such that:



- Il predecessore di 1 e' 0: quindi l'allocazione di Bird assegna a 1 il costo di {1,0}.
- Il predecessore di 2 e' 1: quindi l'allocazione di Bird assegna a 2 il costo di {2,1};
- Il predecessore di 3 e' 1: quindi l'allocazione di Bird assegna a 3 il costo di {1,3}.

$w(\Gamma)=52$

L'allocazione di Bird rispetto a Γ $(x_1, x_2, x_3)=(24, 18, 10)$ sta nel nucleo Core({1,2,3},c).

Note su mcst game

- Si dimostra che dato un mcst game, un’allocazione di Bird sta nel nucleo del mcst ga,e stesso;
- quindi i mcst game sono totalmente bilanciati (cioè bilanciati e per ogni mcst game, tutti i suoi sottogiochi sono anche bilanciati).

Problemi di scheduling

- Rientrano in questa categoria i *sequencing game*, *permutation game*, *assignment game.*
- Sono giochi in cui la funzione caratteristica dipende dalla posizione dei giocatori in una coda;
- i giocatori stessi possono essere visti come venditori della loro posizione iniziali e compratori della loro posizione finale.

Permutation situation <N,A>:

- N={1,...,n} insieme dei giocatori e A matrice dei costi di processo NxN;
- Ogni giocatore ha un lavoro ed una macchina;
- ad ogni macchina è permesso di processare qualsiasi lavoro;
- a nessuna macchina è permesso di processare piu' di un lavoro;
- se il giocatore *i* processa il proprio lavoro sulla macchina del giocatore *j*, il costo di processo è a_{ij} elemento di A di riga *i* e colonna *j*.

Permutation problem.

Problema di Ottimizzazione:

Quale lavoro assegnare a quale macchina per far si' che il costo di processamento sia minimo, ovvero

il risparmio ottenuto rispetto la situazione in cui ogni agente processa il proprio lavoro sulla propria macchina sia massimo?

Permutation game

- Data una situazione di permutazione <N,A>
- Si definisce il permutation game <N,v> tale che

$$v(S)= \sum_{i\in S} a_{ii} - \min_{p\in P_S} \sum_{i\in S} a_{ip(i)}$$

  per ogni $S\subseteq N$, con $S\neq\emptyset$ (naturalmente per definizione $v(\emptyset)=0$) e P_S e' l'insieme di tutte le permutazioni su S.
- Il numero v(S) denota il massimo risparmio che una coalizione S può ottenere dal processare i propri lavori in accordo ad un programma ottimale comparato alla situazione in cui ciascun giocatore processa il proprio lavoro sulla propria macchina.

Example: Sia N={1,2,3} l'iniseme dei giocatori e sia

$$A=\begin{pmatrix} 8 & 4 & 2 \\ 2 & 4 & 10 \\ 5 & 6 & 10 \end{pmatrix}$$

la matrice dei costi. Allora il corrispondente Permutation game è dato da

S	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	{1,2,3}
v(S)	0	0	0	6	11	0	12
Permutazione ottima	p*=(1)	p*=(2)	p*=(3)	p*=(2,1)	p*=(3,1)	p*=(2,3)	p*=(3,1,2)
	8-8=0	4-4=0	10-10=0	12-6=6	18-7=11	14-14=0	22-10=12

Si noti che v({1,2,3})-v({1,3})=1<6=v({1,2})-v({1}) che implica che i permutation games non sono convessi.

Note sui Permutation game

- Si può dimostrare che i permutation games sono totalmente bilanciati.
- Una classe particolare di Permutation games sono gli *assignment game* introdotti da Shapley e Shubik nel 1971. Tali giochi sono ispirati a mercati two-sided in cui merci indivisibili sono scambiate con denaro (modello utilizzato per mercato privato di auto usate, aste ecc.).