Università Roma Tre – Facoltà di Scienze M.F.N – Corso di Laurea in Matematica

a.a. 2001/2002

Barriere assorbenti nelle catene di Markov e una loro applicazione al web

Giulio Simeone

1

Sommario

- Descrizione e formalizzazione del problema di ottimizzazione
- Utilizzo delle catene di Markov per la soluzione del problema
- Applicazione di un algoritmo efficiente per la ricerca della soluzione

2

Descrizione del problema

- Web: è una rete di "pagine" collegate fra loro da *link ipertestuali*
- È possibile "vendere" la collocazione di *banner pubblicitari* sulle pagine del sito, pagati in proporzione al numero di accessi compiuti dagli utenti
- Vogliamo individuare la collocazione ottimale dei banner in modo da *massimizzare il profitto*

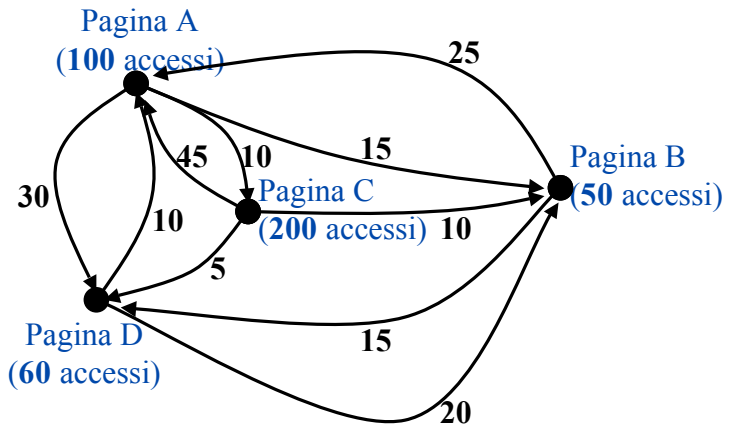
3

Dati del problema

- L'attività degli utenti nella lettura delle pagine di un sito web viene *tracciato* mediante appositi *file di log*:
  - **Access log**: ci informa sul numero di utenti che ha letto una determinata pagina
  - **Referer log**: ci informa sui **link** seguiti dagli utenti all'interno del sito (ci dice da quale pagina "proviene" l'utente)
- I due file ci permettono di costruire un **grafo pesato** (o una *rete di flusso*) che rappresenta l'attività sul sito web

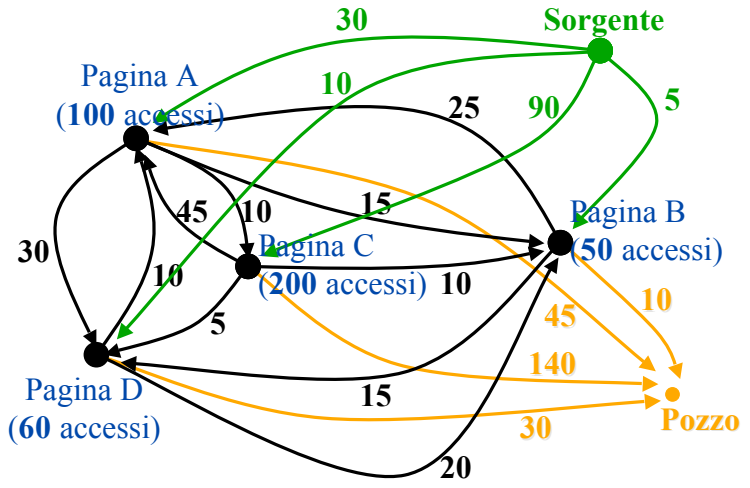
4

Rappresentazione del problema



5

Rappresentazione del problema



6

Il problema di ottimizzazione

- Inserire un banner in una pagina del sito corrisponde a:
  - Aggiungere una pagina (esterna), ossia **aggiungere un vertice** al grafo
  - Aggiungere un link dalla pagina in cui è presente il banner a quella pubblicizzata dal banner, ossia **aggiungere uno spigolo** al grafo
- Dobbiamo trovare la collocazione dei banner tale da rendere massimo il profitto, funzione della **probabilità che il visitatore termini la visita del sito su una delle pagine pubblicizzate** e dei **coefficienti di redditività** dei singoli banner

7

Il problema di ottimizzazione

- Per ciascuna disposizione dei banner nelle pagine del sito il grafo verrà **ampliato** aggiungendo un certo numero di spigoli e di vertici
- Il processo di calcolo della disposizione ottima del banner, viene reso più arduo dal fatto che **i banner da collocare sono numerosi** e l'inserimento di un banner influenza la collocazione di tutti gli altri
- Se la quantità dei banner e delle pagine del sito sono elevate, allora **le possibili disposizioni sono molto numerose**
- Dunque il processo di **ricerca esaustiva** della soluzione ottima, su un grafo di grandi dimensioni, **non è attuabile**

8

La funzione obiettivo

- L'obiettivo è la massimizzazione di una funzione **profitto**:
  - Il profitto per una determinata disposizione di banner è dato dalla somma dei guadagni stimati per ciascun banner
  - Tale guadagno è funzione della **probabilità** con cui un visitatore del sito selezionerà il banner e del **coefficiente di redditività** dello stesso banner

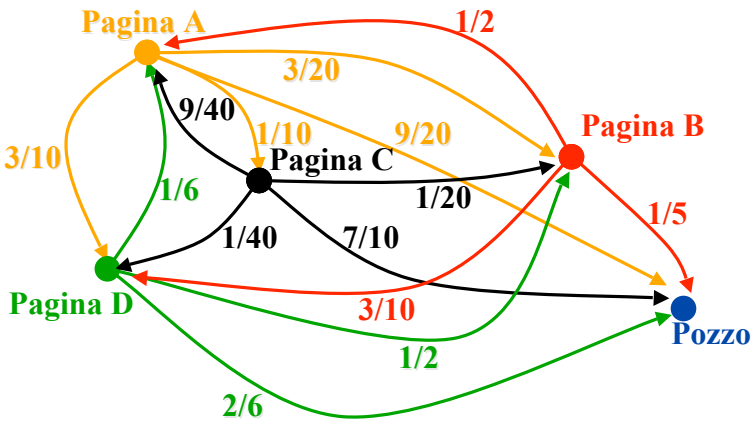
9

Probabilità di transizione

- È necessario quindi trasformare i **valori assoluti** del numero di utenti che percorrono determinati spigoli del grafo, in **probabilità di transizione** da un vertice ad un altro
- Inizialmente le probabilità associate ad ogni spigolo del grafo sono calcolate normalizzando i valori assoluti estratti dal **referer log**
- Successivamente, dopo aver ampliato il grafo con l'aggiunta dei banner, per ogni disposizione, vengono ricalcolate le probabilità assegnate ad ogni spigolo, tenendo anche conto dell'**indice di attrattività** assegnato ad ogni banner

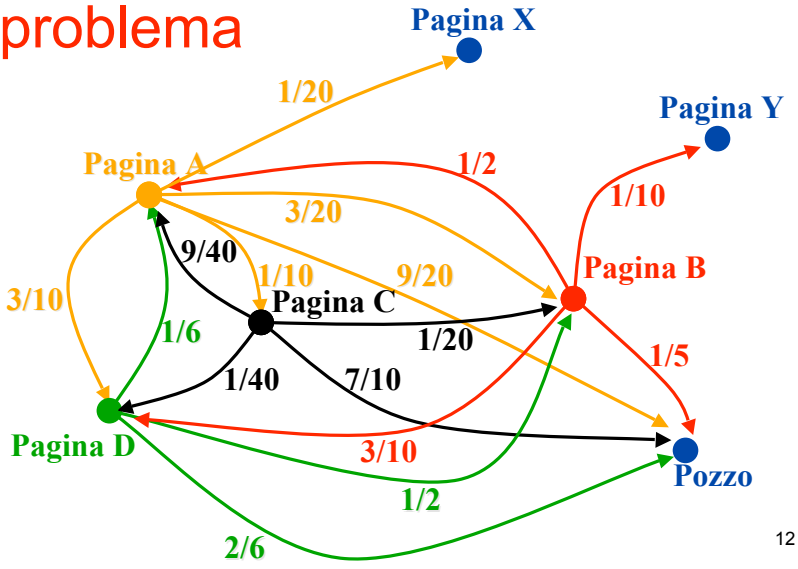
10

Rappresentazione del problema



11

Rappresentazione del problema



12

Catene di Markov

- Il problema si presta ad essere interpretato come una **catena di Markov**
- Infatti il grafo con le probabilità associate agli spigoli può essere letto come un **processo stocastico** con uno **spazio degli stati finito**.
- Le probabilità con cui il sistema si sposta da uno stato all'altro sono **indipendenti dal tempo** (e quindi dalla "storia" delle transizioni precedenti)

13

Matrice di transizione

- La **matrice di transizione** P di una catena di Markov descrive le probabilità con cui il sistema passo da uno stato i al tempo t ad uno stato j al tempo t+1
- Si definisce la **matrice potenziale** G

$$G = \sum_{n \geq 0}^{\infty} P^n$$

- L'elemento $g_{i,j}$ è il numero (finito o infinito) atteso di visite in j se la catena parte da i.

14

Barriere assorbenti

- Sono stati del processo stocastico da cui non è possibile uscire
- I vertici aggiunti al grafo (*pozzo di disconnessione*, *pagine pubblicitarie*) sono **barriere assorbenti** del processo stocastico da noi rappresentato
- Dalla teoria delle catene di Markov sappiamo che il sistema **prima o poi deve finire su una barriera assorbente**
- Grazie ad alcuni teoremi è possibile calcolare la probabilità con cui il sistema termina su una **determinata** barriera assorbente

15

Probabilità di assorbimento

- **Teorema**: Sia P la matrice di transizione di una catena di Markov e G la sua matrice potenziale:

$$P = \begin{pmatrix} I & 0 \\ B & Q \end{pmatrix} \qquad G = \begin{pmatrix} E & 0 \\ F & U \end{pmatrix}$$

- Allora la probabilità che il sistema sia assorbito dallo stato j partendo dallo stato i è data da

$$(UB)_{i,j}$$

16

Algoritmo di ottimizzazione

- Assegnati i banner e gli indici di attrattività, l'algoritmo deve **calcolare le probabilità di assorbimento dei banner** per ognuna delle disposizioni possibili
- Tale calcolo sarebbe eccessivamente oneroso su grafi di grandi dimensioni
- Dunque adottiamo un approccio approssimante noto come **Algoritmo Old Bachelor** (migliore di *Simulated Annealing* e *Threshold Acceptance*)

17

Algoritmi di ricerca della soluzione ottima

- Sia S l'insieme delle soluzioni: vogliamo trovare la soluzione ottima, che minimizza la funzione
$$f(s_i) \qquad s_i \in S$$
- L'algoritmo genera una soluzione di partenza s_0 e fissa una soglia T: le altre soluzioni $s_1,\ldots,s_n$ vengono generate una dopo l'altra a partire dalla precedente tramite una procedura prefissata, che accetta la nuova soluzione se è migliore della precedente o se rientra nella soglia T
- Ad esempio, ogni disposizione s_{i+1} di link nelle pagine del sito viene generata a partire da quella precedente s_i scegliendola a caso in modo tale che le due soluzioni differiscano solo per la collocazione di una pagina

18

Simulated annealing e Threshold acceptance

Algoritmo SA

1) For i=1, …, M
2) sceglie casualmente una soluzione s' vicina a s_i
3) se $f(s') < f(s_i)$ allora $s_{i+1}=s'$
4) altrimenti $s_{i+1}=s'$ con probabilità $\exp((f(s_i)-f(s')/T_i)$
5) $T_i = \text{next}(T_i)$
6) End
7) Restituisce la soluzione s_i per la quale $f(s_i)$ è minima

Algoritmo TA

1) For i=1, …, M
2) sceglie casualmente una soluzione s' vicina a s_i
3) se $f(s') < f(s_i)+T_i$ allora $s_{i+1}=s'$
4) altrimenti $s_{i+1}=s_i$
5) $T_i = \text{next}(T_i)$
6) End
7) Restituisce la soluzione s_i per la quale $f(s_i)$ è minima

19

Algoritmo Old Bachelor

1) Sceglie una soluzione iniziale s_0
2) Sceglie la soglia iniziale T_0
3) For i=0,…,M
4) sceglie casualmente una soluzione s' vicina a s_i
5) se $f(s') < f(s_i)+T_i$ allora $s_{i+1}=s'$ e $T_{i+1}=T_i\text{-decr}(T_i)$
6) altrimenti $s_{i+1}= s_i$ e $T_{i+1} = T_i + \text{incr}(T_i)$
7) End
8) Restituisce la soluzione s_i per la quale $f(s_i)$ è minima

20