Indice

Sommario

In questa tesi si sono studiate le reti di code a capacità limitata con struttura ad albero. Se ne dimostra la reversibilità e la forma prodotto della distribuzione all'equilibrio, sotto le ipotesi che le singole code siano modellabili mediante catene di Markov, il routing sia di Markov, i collegamenti siano bidirezionali. Inoltre per reti aperte si suppone che gli utenti possano entrare (dall'esterno) o uscire (verso l'esterno) attraverso una sola coda. Per reti a struttura stellare composta (albero a tre livelli) è stato implementato un codice che calcola le distribuzioni di probabilità marginali e la costante di normalizzazione della rete, utilizzando l'algoritmo di Buzen [1] per reti a capacità limitata con distribuzione all'equilibrio in forma di prodotto.

Introduzione

L'oggetto di questa tesi è lo studio di una classe di modelli di sistemi dinamici ad eventi discreti : le reti di code. Tali modelli rivestono una notevole importanza in svariati campi applicativi che vanno dall'informatica, all'industria manufatturiera, alle comunicazioni, per la lora rapidità e semplictà di utilizzo che li rende competitivi con i modelli simulativi più precisi ma più complicati e lenti. In questo lavoro si è analizzata una particolare classe di reti di code : le reti di code con struttura ad albero a collegamenti bidirezionali. Lo studio è stato svolto anche nel caso in cui le stazioni siano a capacità limitata. Per le reti di code con struttura a stella composta è stato sviluppato un codice che calcola le distribuzioni marginali di tutte le code e la costante di normalizzazione.

Il primo modello di reti di code fu proposto da Jacskson [6], dove si analizzano reti di code aperte a buffer illimitato, in cui gli utenti arrivano dall'esterno secondo processi di Poisson, e i tempi di servizio sono distribuiti esponenzialmente.

Gordon e Newell in [3] estesero i risultati di Jackson considerando reti di code chiuse . Sono state fatte varie estensioni di queste modelli, in particolare le code a buffer limitato sono state studiate da Hordijk - Van Dijf [5] e da Yao - Buzacott in [17]. Il concetto di reversibilità fu introdotto da Kelly, che lo studiò diffusamente in [7]. Kingman in [10] introdusse i processi migratori reversibili, ripresi da Pollet in [13]. Il concetto di quasi reversibilità fu introdotto algebricamente da Muntz in [12] e probabilisticamente da Kelly in [7]. In letteratura non si trova uno studio analitico generale di reti di code a capacità limitata con truttura ad albero. Yao - Buzacott [15] trattano le reti di code (a capacità limitata) cosiddette a stella semplice (albero a due livelli).

In questa tesi si considera il caso più generale di rete a capacità limitata con struttura ad albero qualsiasi. Se ne dimostra la reversibilità e la forma prodotto della distribuzione all'equilibrio, sotto di modellabilità mediante catene di Markov, collegamenti bidirezionali, inoltre per reti aperte si suppone che gli utenti possano entrare (dall'esterno) o uscire (verso l'esterno) attraverso una sola coda.

Nel capitolo 1 viene data una sintesi delle catene di Markov limitatamente a quanti interessa i capitoli successivi e vengono introdotte le code e le reti di code.

Nel capitolo 2 viene studiata la reversibilità e vengono forniti criteri di necessità e sufficienza.

Nel capitolo 3 viene esaminata la proprietà di quasi reversibilità e si prende in considerazione il collegamento di reti di code quasi reversibili studiandone la reversibilità e la forma prodotto della distribuzione all'equilibrio.

Nel capitolo 4 si prendono in considerazione le reti di code a capacità limitata con struttura da albero.

Nel capitolo 5 viene studiato una particolare classe di catene di Markov : i processi migratori. Viene ampiamente discussa la tecnica di interconnesione di catene di Markov reversibili. Si dimostra la reversibilità e la forma prodotto delle reti con struttura a stella composta utilizzando anche il teorema di Pollet.

Nel capitolo 6 viene esposto un algoritmo che permette il calcolo della costante di normalizzazione per reti di code a capacità limitata.

In appendice si riporta il codice sviluppato per il calcolo delle distribuzioni di probabilità marginali e della costante di normalizzazione per reti a struttura stellare composta.

Capitolo 1

PROCESSI DI MARKOV

1.1 Introduzione ai processi di Markov

Si consideri un processo aleatorio $x(t)$ che assume valori in un insieme $\mathcal{X}$ per $t \in \mathcal{T}$. Si dice che $x(t)$ è un un processo aleatorio di Markov se $\forall t, s \in \mathcal{T}$ e $\forall A \subset \mathcal{X}$ gode dell proprietà :

$$P[(x(s) : s \geq t) \in A | x(s) : s \leq t] = P[(x(s) : s \geq t) \in A | x(t)] \tag{1.1}$$

In un processo di Markov la probabilità di un evento futuro, condizionato dagli eventi passati e dal presente è indipendente dagli eventi passati e dipende solamente dall'evento presente. In altri termini lo stato presente, $x(t)$, contine tutta l'informazione riguardante la passata evoluzione del processo, che risulta necessaria per prevedere la sua evoluzione futura.

In particolare, se $\mathcal{X}$ (insieme deli stati) è numerabile , $x(t)$ assume il nome di catena di Markov, a tempo continuo se $t \in \mathcal{T} \subseteq \Re$ (insieme dei numeri reali), a tempo discreto se $t \in \mathcal{T} \subseteq Z$ (insieme dei numeri interi).

Definizione : Una catena di Markov a tempo continuo si dice stazionaria in distribuzione di ordine n se $\forall t_1, ..., t_n, \tau \in \Re$ e $j_1, ..., j_n \in \mathcal{X}$ si ha:

$$P[x(t_1) = j_1, ..., x(t_n) = j_n] = P[x(t_1 + \tau) = j_1, ..., x(t_n + \tau) = j_n] \tag{1.2}$$

In particolare se una catena di Markov a tempo continuo è stazionaria in distribuzione di ordine n si ha che $\forall j, k \in \mathcal{X}$ e $\forall t, \tau \in \Re :$ $P[x(t + \tau) = j | x(t) = k]$ è indipendente da t e dipende solamente da τ (stazionarietà in distribuzione di ordine due).

Definizione : Una catena di Markov, a tempo continuo, si dice omogenea se $\forall t, \tau \in \Re, \tau \geq 0$ e $\forall j, k \in \mathcal{X}$ si ha che : $P[x(t+\tau) = j | x(t) = k]$ è indipendente da t.

Definizione : Data una catena di Markov, a tempo continuo,omogenea si definisce tasso di transizione la seguente quantità :

$$q(i,j) = \lim_{\tau \to 0^+} \frac{P[x(t+\tau) = j | x(t) = i]}{\tau} \tag{1.3}$$

$$\forall i \neq j \in \mathcal{X} \; e \; \forall t, \tau \in \Re$$

Si pone inoltre $q(j) = 0 \; \forall j \in \mathcal{X}$ e si impone che il limite dell'equazione (1.3) sia finito. Si noti che la definizione di tasso di transizione é indipendente da t poiché il processo é omogeneo.

Definizione : Data una catena di Markov, a tempo continuo, ed omogenea si definisce rate matrix Q la matrice che ha per componenti i tassi di transizione $q(i,j)$. La rate matrix Q è definibile poiché l'insieme degli stati $\mathcal{X}$ è numerabile .

1.2 Evoluzione di una catena di Markov

Sia $\mathcal{X}$ un insieme numerabile, Q una rate matrix su $\mathcal{X}$ e $\pi_0(i) \in \mathcal{X}$ la probabilità che $x(0) = i$. La catena di Markov associata a Q inizialmente ($t = 0$) assume valore $x(0) = i$ con probabilià $\pi_0(i)$. Dopodichè $x(t)$ mantiene quel valore per un tempo τ distribuito esponenzialmente con parametro $q(i) = \sum_{j \in \mathcal{X}} q(i,j)$ (somma della riga i-esima della matrice Q). All'istante τ il processo passa nello stato $x(\tau)$ con probabilità :

$$p(i,j) = \frac{q(i,j)}{q(i)} = \frac{q(i,j)}{\sum_{j \in \mathcal{X}} q(i,j)} \, 1(i \neq j)$$

Le $p(i,j)$ sono dette probabilità di transizione mentre la funzione $1(C)$ vale uno se C è vera, zero altrimenti. A questo punto l'evoluzione ricomincia da $x(\tau)$ indipendentemente da quanto successo prima (cioè come se τ fosse l'istante iniziale).

Il processo così definito è una catena di Markov omogenea. La verifica si ottiene direttamente sfruttando la cosiddetta mancanza di memoria della variabile aleatoria esponenziale unilatera , e dal fatto che ,in ogni istante di transizione , il nuovo valore viene scelto con una probabilità che dipende solo dallo stato precedente.

Si può dimostrare (Grimmett-Stirzaker) [4] che, data una catena di Markov con rate matrix Q su $\mathcal{X}$, la sua evoluzione è quella precedentemente descritta.

1.3 Distribuzione invariante, limite e all'equilibrio

Definizione : Una catena di Markov, a tempo continuo, si dice regolare se compie un numero finito di transizioni in un tempo finito.

L'ipotesi di regolarità è necessaria per stabilire una corrispondenza biunivoca tra la catena di Markov a tempo continuo e la sua rate matrix. Nel caso infatti di infinite transizioni in un tempo finito la catena potrebbe essere reinizializzata in un qualunque modo.

Definizione : Si definiscono inoltre le probabilità :

$$P_t(i,j) = P[x(t) = j | x(0) = i] \ ; \ \forall t > 0, \forall i, j \in \mathcal{X}$$

$$\pi_t(i) = P[x(t) = i] \ ; \ \forall i \in \mathcal{X}$$

La probabilità $\pi_t(i)$ è detta distribuzione invariante se èindipendente dalla variabile temporale t.

Il seguente teorema permette di calcolare il vettore riga π_t :

TEOREMA (1.1)

Siano $(P_t : t \geq 0)$ le probabilità di transizione di una catena di Markov a tempo continuo regolare con rate matrix Q, si hanno i seguenti risultati :

a) $P_{t+s} = P_t P_s \quad \forall s, t \geq 0$

b) $d(P_t)/dt = QP_t = P_tQ \qquad \forall t \geq 0$

c) Se Q é uniforme :

$$P_t = e^{Qt} = \sum_{n=0}^{+\infty} \frac{Q^n t^n}{n!} \; t \geq 0$$

d) La distribuzione di probabilità π_t non dipende da t se e solo se π_0 é soluzione delle equazioni $\pi Q = 0$, cioè se e solo se $\forall i \in \mathcal{X}$ si ha :

$$\pi(i) \sum_{j \in \mathcal{X}} q(i,j) = \sum_{j \in \mathcal{X}} q(j,i)$$

dimostrazione :

Si considera per semplicità il caso in cui $\mathcal{X}$ sia un insieme finito.

a):

$$P[x_{t+s} = j | x_0 = i] = \sum_k P[x_{t+s} = j, x_t = k | x_0 = i] =$$

$$= \sum_k P[x_{t+s} = j | x_t = k, x_0 = i] P[x_t = k | x_0 = i] =$$

$$= \sum_k P[x_{t+s} = j | x_t = k] P[x_t = k | x_0 = i] =$$

$$= \sum_k P[x_s = j | x_t = k] P[x_t = k | x_0 = i] =$$

$$= \sum_k P_s(k,j) P_t(i,k)$$

Quindi $P_{t+s} = P_t P_s$.

b):

Si verifica inizialmente che $d(P_t)/dt = Q$ per $t = 0$. Assumendo che τ sia il primo istante di transizione si ha :

$$P[x_t = j | x_0 = i] = P[\tau \leq t | x_0 = i] P[x_\tau = j | x_0 = i] =$$

$$= (1 - e^{-t \sum_k q(i,k)}) \frac{q(i,j)}{\sum_k q(i,k)} =$$

sviluppando in serie di $e^{-\sum_k q(i,k)}$ e indicando con $\sigma(t)$ una funzione infinitesima per $t \longrightarrow 0$ con ordine di infinitesimo superiore al primo :

$$= q((i,j)t + \sigma(t)$$

Quindi $q(i,j) = d(P_0(i,j))/dt$. In base a quanto visto $q(i,j)t$ è a meno di infinitesimi di ordine superiore, la probabilità di passare da i a j in un tempo t, e $1 - \sum_k q(i,j)$ è la probabilità di essere rimasti nello stato i dopo un tempo t (a parte infinitesimi di ordine superiore). si ha quindi (I matrice identità) :

$$P_{t+\epsilon} = P_t P_\epsilon = P_t[I + Q\epsilon + \sigma(\epsilon)] == P_\epsilon P_t = [I + Q + \sigma(\epsilon)]P_t$$

da cui, sottraendo P_t e dividendo per ϵ si ottiene quanto si vuole dimostrare :

$$\frac{dP_t}{dt} = QP_t = P_t Q$$

facendo tendere ϵ a zero.

c):

La si può verificare derivando il suo termine destro e vedendo che risolve l'equazione del punto b). L'ipotesi di uniformitá della matrice Q serve a garantire la convergenza della serie.

d):

Se $\pi_0 Q = 0$, allora , poiché :

$$\pi_t = \pi_0 P_t \pi_0 \sum_{n=0}^{\infty} \frac{Q^n t^n}{n!}$$

per il teorema della probabilità totale , si ha che $\pi_t = \pi_0$.

Viceversa si suppone che $\pi_t = \pi_0 P_t = \pi_0$, allora derivando per $t = 0$ si ha proprio la $\pi_0 Q = 0$.

Si verifica per ultimo la stazionarietà di $x(t)$.

Per ogni insieme di realizzazioni A , abbiamo:

$$P[(x_{t+s} : s \geq 0) \in A] = \sum_i P[x_t = i] P[(x_{t+s} : s \geq 0) \in A | x_t = i]$$

$$\sum_i \pi_t(i) P[(x_s : s \geq 0) \in A | x_0 = i]$$

Dunque se $\pi_t = \pi_0$, abbiamo che $P[(x_{t+s} : s \geq 0) \in A]$ non dipende da t , dunque la stazionarietà . (fine dimostrazione)

Definizione : una catena di Markov a tempo continuo si dice irriducibile su $\mathcal{X}$ (spazio degli stati) quando $\forall t \in \Re$; $j, k \in \mathcal{X}$ esiste $\tau \in \Re$ tale che:

$$P[x_{t+\tau} = k | x_t = j] \not= 0$$

cioè quando ogni stato comunica direttamente o inderettamente con tutti gli altri in un tempo finito.

Definizione : si dice distribuzione limite di una catena di Markov a tempo continuo la quantità :

$$\lim_{t \to \infty} P_t(i,j) \; ; \; i, j \in \mathcal{X}$$

TEOREMA (1.2):

Se una catena di Markov a tempo continuo regolare e irriducibile allora ha , se esiste , una sola istribuzione invariante. In caso di esistenza tale distribuzione coincide con la distribuzione limite cioè :

$$\lim_{t \to \infty} P_t(i,j) = \pi_t(j) \; ; \; i, j \in \mathcal{X}$$

Definizione : si dice distribuzione all'eqilibrio di una catena di Markov , a tempo , continuo un insieme di numeri positivi $\{\pi(i) i \in \mathcal{X}\}$ che soddisfano alle seguenti proprietà :

$$\pi(i) > 0 \quad \forall i \in \mathcal{X}$$

$$\sum_{i \in \mathcal{X}} \pi(i) = 1$$

$$\pi(i) \sum_{j \in \mathcal{X}} q(i,j) = \sum_{j \in \mathcal{X}} \pi(j) q(j,i) \quad \forall i \in \mathcal{X} \tag{1.4}$$

L'equazione (1.4) è detta equazione di bilancio all'equilibrio e stabilisce che il flusso di tutte le transizioni entranti in i è uguale al flusso di tutte le transizione uscenti da i. Dalla definizione di distribuzione all' equilibrio si deduce che se esiste essa è unica, dal teorema (1.1) (se la catena di Markov è regolare) si deduce che, sempre se esiste, essa coincide con la distribuzione invariante (da cui la stazionarietà).

Per il teorema (1.2) (sotto ipotesi di regolarità e irriducibilità) la distribuzione invariante coincide con la distribuzione limite. Quindi per catene di Markov a tempo continuo regolari irriducibili se esiste la distribuzione all' equilibrio essa è anche la distribuzione invariante e limite.

Nel corso della tesi si considereranno solo catene di Markov a tempo continuo regolari irriducibili.

Le catene di Markov sono utilizzate per modellare , sotto opportune ipotesi, sistemi dinamici come le code singole e le reti di code. Nei prossimi paragrafi si introdurranno tali sistemi dinamici.

1.4 Coda singola

Si parla di coda singola ogni qualvolta si è in presenza di utenti (possono essere anche oggetti) richiedenti un servizio, erogato da una stazione fornitrice di tale servizio. Se l'utente che si presenta alla stazione la trova occupata , si mette in coda e attende che il servizio si liberi. Una volta completato il proprio servizio l'utente esce dalla stazione di servizio.

Una coda è formalmente definita quando si specificano :

- il processo degli arrivi (la statistica secondo cui gli utenti giungono alla stazione erogatrice di servizio)
- la distribuzione di probabilità dei tempi di servizio (statistica secondo cui gli utenti escono dalla stazione servente).
- disciplina di servizio (priorità seguita)

Si è soliti rappresentare una coda con la notazione :

A-B-s-disciplina

dove :

- la lettera A indica la modalità di arrivo degli utenti nella coda e poò assumere i valori : M (se i tempi di interarrivo sono esponenziali e indipendenti),G (se il processo di arrivo è generale),GI (se i tempi di interarrivo sono indipendenti e indenticamente distribuiti con distribuzione generale),D (se i tempi di interarrivo sono costanti e uguali tra loro.

- la lettera B indica la distribuzione dei tempi di servizio e può assumere i valori di A con lo stesso significato.

- la lettera s indica il numero di centri di servizio che operano in parallelo nella stazione di servizio

- la disciplina di servizio può essere di vari tipi, fra le più comuni si elencano :

a) **FCFS :** (first come first served) quando viene servito per primo l'utente che ha il più lungo tempo di permanenza nella coda.

b) **LCLS :** (last come last served) quando viene servito per primo l'utente che ha il più breve tempo di permanenza nella coda.

c) **PS :** (processor shared) quando il tempo di servizio viene equamente suddiviso fra tutti gli utenti presenti nella coda .

d) **IS :** (infinite servers) quando la stazione ha infiniti centri erogatori di servizio .

È possibile descrivere l'evoluzione di molte code utilizzando il modello delle catena di Markov. Dalla definizione di catena di Markov, e ricordando che solamente la variabile esponenziale unilatera gode della proprietà di 'assenza di memoria' (detta anche proprietà di Markov), si può verificare che è possibile modellare una coda mediante una catena di Markov se e solamente se i processi degli arrivi e delle partenze hanno entrambi distribuzione esponenziale.

Esempio (coda M-M-1 a capacità limitata) :

Nella coda M-M-1 (discipline FCFS, LCLS, PS) i tempi di interarrivo sono variabili aleatorie esponenzialia unilatere indipendenti ed identicamente distribuite con parametro λ (il processo degli arrivi è un processo di Poisson omogeneo). I tempi di servizio sono anch'essi i.i.d (indipendenti e identicamente distribuiti) con parametro μ. La coda è supposta a capacità limitata, cioè non può contenere più di K utenti. Si vuole verificare che tale coda è modellabile mediante una una catena di Markov con sazio degli stati $\mathcal{X} = \{0, 1, ..., K\}$ (numero degli utenti in coda).

Il tempo di permanenza τ nello stato n $(0 < n < K)$ è una variabile aleatoria esponenziale con parametro $\lambda + \mu$, infatti, detto t_λ il tempo di interarrivo e t_μ il tempo di servizio si ha che $(t \geq 0)$:

$$P[\tau > t] = P[t_\lambda > t, t_\mu > t] =$$

Poichè t_λ e t_μ sono supposte indipendenti

$$= P[t_\lambda > t]P[t_\mu > t] = e^{-(\lambda+\mu)t}$$

Quindi τ ha distribuzione di probabilità $(t \geq 0)$:

$$F_\tau = 1 - e^{-(\lambda+\mu)t}$$

(distribuzione di una variabile aleatoria esponenziale di parametro $\lambda + \mu$).

Dopo il tempo τ lo stato compie la transizione $n \rightarrow n+1$ con probabilità data da $(f_{t_\mu}(a)$ densità di $t_\mu)$:

$$P[t_\lambda < t_\mu] = \int_{-\infty}^{+\infty} P[t_\lambda < a | t_\mu = a] f_{t_\mu}(a)\, da =$$

dove $f_{t_\mu}(a)$ èla densità di t_μ

$$= \int_{-\infty}^{+\infty} P[t_\lambda < a] f_{t_\mu}(a)\, da =$$

$$= \int_{0}^{+\infty} (1 - e^{-\lambda a}) \mu e^{-\mu a}\, da = \frac{\lambda}{\lambda + \mu}$$

La transizione $n \to n-1$ avviene con probabilità

$$P[t_\lambda > t_\mu] = 1 - P[t_\lambda < t_\mu] = \frac{\mu}{\lambda + \mu}$$

Per $n = 0$ ($n = K$) il tempo di permanenza τ è una variabile aleatoria esponenziale di parametro λ (μ) e la probabilità di passare , dopo il tempo τ, nello stato $n = 1$ ($n = K - 1$) è unitaria.

Quindi l'evoluzione della lunghezza di una coda M-M-1 è descrivibile mediante una catena di Markov definita su $\mathcal{X} = 0, 1, ..., K$ con tassi di transizioni :

$$q(n, n+1) = \lambda \;\; ; \;\; 0 \leq n < K$$

$$q(n, n-1) = \mu \;\; ; \;\; 1 \leq n < K$$

$$q(n, l) = 0 \; ; \; \mathit{altrimenti}$$

Per ricavare $\pi(i)$ $i \in \mathcal{X}$ si può utilizzare l'equazione di bilancio all'equlilibrio :

$$\pi(i) \sum_{j \in \mathcal{X}} q(i, j) = \sum_{j \in \mathcal{X}} \pi(j) q(j, i)$$

che fornisce, nel caso specifico, per $0 \leq i \leq K$:

$$\pi(i) q(i, i+1) = \pi(i+1) q(i+1, i)$$

cioè :

$$\pi(i) \lambda = \pi(i+1) \mu$$

da cui :

$$\pi(i) = c \left(\frac{\lambda}{\mu} \right)^i$$

dove c si determina imponendo $\sum_{i \in \mathcal{X}} \pi(i) = 1$ ($c = \frac{1-q}{1-q^{K+1}}$).

Dunque esiste la distribuzione all'equilibrio , e giacchè la catena di Markov associata alla coda M-M-1 è regolore e irriducibile, essa è unica e coincide con la distribuzione invariante e limite.

1.5 Reti di code

Si parla di reti di code quando si è in presenza di un insieme di code collegate fra loro.

Una rete di code è univocamente definita quando vengono fissati :

- l'insieme delle code, ognuno con la sua descrizione formale

- le probabilità di transizione di un utente da una coda all'altra

- i processi di arrivo dall'esterno

Definizione : una rete di code è detta aperta se gli utenti arrivano dall'esterno ed escono verso l'esterno ,si dice invece chiusa quando nessun utente può entrare o uscire dalla rete (il numero degli utenti nella rete è costante) .

Se un utente richiede un determinato servizio ad una stazione satura (numero di utenti nel sistema uguale alla capacità) viene bloccato (cioè non viene accettato). In questa tesi si farà sempre riferimento al bloccagio "block and hold" : l'utente richiedente un determinato servizio ad una stazione satura viene bloccato e resta nella stazione in cui si trova.

Definizione : data una rete di code si nomina routing la legge secondo cui un utenti si muove attraverso i vari centri del sistema . Questa legge è determinata dalle probabilità che un utente, terminato il servizio in un centro, vada in una delle altre code o si diriga verso l'esterno.

Definizione : un routing si dice di Markov (o senza memoria) se le decisioni che riguardano il routing sono tutte indipendenti tra loro ed indipendenti dalla passata evoluzione del processo che descrive la rete.

Definizione : si dice rete multiclasse una rete nella quale sono presenti utenti di più classi (il routing della rete e i tempi di servizio possono dipendere dalla classe in considerazione). In questa sede ci si occupererà di reti di code con una sola classe di utenti.

1.6 Applicazioni delle reti di code

Le reti di code rivestono una notevole importanza in una vasta classe di campi applicativi, che vanno dall'informatica, all'industria alla biologia, per la loro rapidità e semplicità d'uso, che li rende competitivi con i modelli simulativi più precisi ma notevolmente più lenti e complicati.

La tendenza è quella di utilizzare i modelli a reti di code per una fase di studi preliminare, e successivamente, per uno studio maggiormente dettagliato e approfondito, utilizzare dei modelli simulativi.

Le applicazioni di maggior interesse sono nel campo delle comunicazioni (trasmissione dati) e nell'industria manufatturiera. Ci si occupererà in particolar modo, della modellizzazione dei cosiddetti sistemi F.M.S (flexible manufactering systems)

Un F.M.S è un insieme di macchine utensili che possono lavorare su una varietà di componenti (famiglie di pezzi) in maniera non presidiato (senza controllo umano), con un sistema di gestione automatizzato delle funzioni ausiliarie (carico e scarico delle macchine, gestione dei pezzi) ed inoltre con distribuzione e circolazione dei pezzi (semilavorati o prodotti finiti) senza presidio.

Tale sistema produttivo è adottato in particolar modo nelle piccole e medie industrie manufatturiere, nelle quali il contenimento dei costi si scontra con la necessità di un ventaglio abbastanza vasto di manufatti, con lotti prodotti in piccola quantità .

Capitolo 2

REVERSIBILITÀ

Prima di passare ad esempi di reti di code è necessario introdurre il concetti di reversibilità delle catene di Markov, molto utile per ottenere la distribuzione all'equilibrio.

2.1 Teoremi sulla reversibilità

Euristicamente parlando, la descrizione statistica completa di alcuni processi aleatori non cambia se la variabile temporale t viene 'invertita' (cambiata di segno),in questo caso si parla di reversibilità . Il concetto di reversibilità risulta molto utile nella determinazione della distribuzione all'equilibrio di particolari reti di code. Si fornisce ora una definizione formale di reversibilità di un processo aleatorio.

Definizione : un processo aleatorio x(t) a tempo continuo si dice reversibile se il vettore $(x(\tau - t_1), ..., x(\tau - t_n))$ risulta avere la stessa descrizione statistica congiunta del vettore $(x(t_1), ..., x(t_n))$, cioè se e solo se:

$$P[x(t_1) \leq a_1, ..., x(t_n) \leq a_n] = P[x(\tau - t_1) \leq a_1, ..., x(\tau - t_n) \leq a_n]$$

$$\forall t_1, ..., t_n, a_1, ..., a_n, \tau \in \Re$$

TEOREMA (2.1) :
Un processo aleatorio a tempo continuo reversibile è stazionario.

Dimostrazione :

per definizione di di reversibilità abbiamo :

$$P[x(t_1) \leq a_1, ..., x(t_n) \leq a_n] = P[x(\tau - t_1) \leq a_1, ..., x(\tau - t_n) \leq a_n] =$$

ponendo $\tau = 0$ si ha :

$$P[x(-t_1) \leq a_1, ..., x(-t_n) \leq a_n] =$$

riappliccando la reversibilità

$$P[x(-\tau + t_1) \leq a_1, ..., x(-\tau + t_n) \leq a_n]$$

da cui la conclusione.

Si illustrano di seguito alcune propietà di base delle catene di Markov reversibili.

TEOREMA (2.2) :

Una catena di Markov stazionaria , a tempo continuo, stazionaria è reversibile se e solo se esiste un insieme di numeri positivi $\{\pi(j) : j \in \mathcal{X}\}$ tale che :

$$\sum_{j \in \mathcal{X}} \pi(j) = 1$$

$$\pi(j)q(j,k) = \pi(k)q(k,j) \quad \forall j,k \in \mathcal{X} \tag{2.1}$$

quando tale insieme di numeri $\{\pi(j) : j \in \mathcal{X}\}$ esiste, esso coincide con la distribuzioneall'equilibrio della catena di Markov.

Dimostrazione :

Si supponga che la catena di Markov sia reversibile (quindi stazionaria) e sia $\pi(j) = P[x(t) = j]$ abbiamo :

$$P[x(t) = j, x(t+\tau) = k] = P[x(\tau - t) = j, x(-t) = k] \ ; \ (reversibilit\grave{a})$$

$$P[x(t+\tau) = j, x(t) = k] \ ; \ (stazionariet\grave{a})$$

osservando che :

$$P[x(t) = j, x(t+\tau) = k] = P[x(t+\tau) = k | x(t) = j]$$

e che

$$P[x(t) = k, x(t+\tau) = j] = P[x(t+\tau) = j|x(t) = k]$$

si ha quindi :

$$\pi(j)\frac{P[x(t) = k, x(t+\tau) = j]}{\tau} = \pi(k)\frac{P[x(t+\tau) = j|x(t) = k]}{\tau}$$

passando al limite per $\tau \longrightarrow 0$ si ottiene l'equazione (2.1).
Viceversa, si suppone che esista un insieme di numeri positivi soddisfacenti alle condizione del teorema. Sommando in k l'equazione (2.1) si ottiene l'equazione (1.4), quindi l'insieme $\{\pi(j)\ j \in \mathcal{X}\}$ è la distribuzione all'equilibrio. Si considera ora la catena x(t) nell' intervallo temporale $[-T, T]$. La catena di Markov può trovarsi nello stato j_1 all'istante $t = -T$, rimanervi per un tempo h_1 prima di passare allo stato j_2, e così via, fino a giungere nello stato j_n dove rimane, finchè $t = T$, per un tempo h_n. La densità di probabilità della variabile aleatoria h_i $(i = 1, ..., n)$ è :

$$q(i)e^{-q(i)a_i} \quad ; \quad a_i \in \Re$$

dove $q(i) = \sum_{j \in \mathcal{X}} q(i,j)$.
La probabilità che j_{i+1} sia lo stato successivo a j_i è :

$$\frac{q(i,i+1)}{q(i)} \qquad 1 \leq i \leq n-1$$

e la densità di probabilità del comportamento descritto :

$$\pi(j_1)e^{-q(j_1)a_1}q(j_1,j_2)e^{-q(j_2)a_2} * ..... * q(j_{n-1},j_n)e^{-q(j_n)a_n} \tag{2.2}$$

Considerando $x(-t)\ t \in [-T, T]$ (catena rovesciata nel tempo) e operando come prima si ha che la densità del comportamento della catena è :

$$\pi(j_n)e^{-q(j_n)a_n}q(j_n,j_{n-1})e^{-q(j_{n-1})a_{n-1}} * ..... * q(j_2,j_1)e^{-q(j_1)a_1} \tag{2.3}$$

La (2.2) e (2.3) sono uguali infatti la (2.1) implica che

$$\pi(j_1)q(j_1,j_2)q(j_2,j_3) * ..... * q(j_{n-1},j_n) =$$

$$= \pi(j_n)q(j_n,j_{n-1})q(j_{n-1},j_{n-2}) * ..... * q(j_2,j_1)$$

questo significa che $x(t)$ ha lo stesso comportamento di $x(-t)$ in $[-T, T]$. Allora $(x(t_1), ...x(t_n))$ ha la stessa densità di probabilità di $(x(-t_1), ...x(-t_n))$ e quindi di $(x(\tau - t_1), ...x(\tau - t_n))$. Da cui discende la reversibilità di $x(t)$.

La quantità $\pi(j)q(k, k)$, considerata dal teorema (2.2), è detta flusso di probabilità dallo stato j allo stato k, e rappresenta la probabilità che la catena di Markov passi dallo stato j allo stato k in un intervallo temporale infinitesimo $(t, t + dt)$ (a parte infinitesimi di ordine superiore al primo).

2.2 Considerazioni sulla reversibilità

I teoremi sulla reversibilità forniscono una via per verificare se un insieme di numeri rappresenta , in realtà , le probabilità $P[x(t) = j])$ $j \in \mathcal{X}$ per una catena di Markov , a tempo continuo regolare irriducubile. Infatti, data una catena di Markov a tempo continuo su $\mathcal{X}$, e dato un insieme di numeri $\pi(j)$ $j \in \mathcal{X}$, (postulati , in questa fase non è nemmeno detto che queste siano delle probabilità), si supponga che tale insieme soddisfi le condizioni :

$$\pi(j)q(j, k) = \pi(k)q(k, j) \quad \forall j, k \in \mathcal{X}$$

$$\sum_{j \in \mathcal{X}} \pi(j) = 1.$$

Si supponga inoltre che la catena di Markov sia regolare e irriducibile. Si ha (poichè $\pi(j)q(j, k) = \pi(k)q(k, j) \Longrightarrow \pi(j) \sum_{k \in \mathcal{X}} q(j, k) = \sum_{k \in \mathcal{X}} \pi(k)q(k, j)$) , per il teorema (1.1), che:

$$\pi(j) = P[x(t) = j]$$

e che $x(t)$ è stazionario.
Per il teorema (1.2) si ha che :

$$\pi(j) = \lim_{t \to \infty} P[x(t) = j | x(0) = k]$$

Per il teorema (2.2) si ha inoltre che la catena di Markov è reversibile.

Il teorema (2.2) fornisce inoltre , in casi semplici,un modo per calcolare l'insieme $\pi(j)\ j \in \mathcal{X}$, tramite l'equazione (2.1), più praticabile di quello fornito dall'equazione di bilancio all'equilibrio (1.4).

In reti di code di una certa complessità il calcolo dell'insieme $\pi(j)$ tramite la (2.1) non è facilmente ottenibile a causa dell' elevato numero di stati. Per questo motivo si sono ricercate ipotesi (quasi reversibilità) per ottenere le $\pi(j)$ in forma analitica (forma prodotto) . Prima di passare alla quasi reversibilità (cap. 3) si danno altri importanti risultati riguardo la reversibilità .

2.3 Altri teoremi sulla reversibilità

A ogni catena di Markov su $\mathcal{X}$ può essere associato un grafo orientato che ha per vertici gli stati $j \in \mathcal{X}$. Nel grafo è presente l'arco (j,k) se e solo se $q(j,k) > 0$. Il seguente teorema afferma che , dato un insieme di stati $B \subset \mathcal{X}$, la somma dei flussi di probabilità da $\{B\}$ a $\{\mathcal{X} - B\}$ è uguale alla somma dei flussi di probabilità da $\{\mathcal{X} - B\}$ a $\{B\}$.

TEOREMA (2.3) :

Data una catena di Markov stazionaria $\forall B \subset \mathcal{X}$ si ha :

$$\sum_{j \in B} \sum_{k \in \mathcal{X}-B} \pi(j)\ q(j,k) = \sum_{j \in B} \sum_{k \in \mathcal{X}-B} \pi(k)\ q(k,j) \tag{2.4}$$

dimostrazione :

dall'equazione di bilancio all'equilibrio

$$\sum_{j \in \mathcal{X}} \pi(j)\ q(j,k) = \sum_{j \in \mathcal{X}} \pi(k)\ q(k,j)$$

sommando in $j \in B$ si ottiene :

$$\sum_{j \in B} \sum_{k \in \mathcal{X}} \pi(j)\ q(j,k) = \sum_{j \in B} \sum_{k \in \mathcal{X}} \pi(k)\ q(k,j)$$

il risultato si ottiene sottraendo l'identità

$$\sum_{j \in B} \sum_{k \in B} \pi(j)\ q(j,k) = \sum_{j \in B} \sum_{k \in B} \pi(k)\ q(k,j)$$

IL teorema (2.3) è utile per dimostrare la seguente condizione sufficiente per la reversibilità .

TEOREMA (2.4) :
Una catena di Markov stazionaria è reversibile se il suo grafo associato è un albero.

Dimostrazione :
Se i vertici j, k non sono collegati la condizione espressa dall'equazione (2.1) è soddisfatta banalmente . Se viceversa j, k sono collegati da un arco ,non orientato poichè il grafo è un albero , ponendo j nell'insieme $\{B\}$ del teorema (2.3) e k nell'insieme $\{\mathcal{X} - B\}$, e assicurandosi che i due insiemi siano collegati solo dall'arco (j, k), si ottiene dall' equazione (2.4) l'equazione (2.1).
(fine dimostrazione)

Si dà ora un importante criterio di reversibilità per una catena di Markov a tempo continuo stazionaria e irriducibile basato solamente sui tassi di transizione.

TEOREMA (2.5) (criterio di Kolmogorv) :
Una catena di Markov a tempo continuo irriducibile e stazionaria è reversibile se e solo se per ogni sequenza finita di stati $j_1, ..., j_n$ si ha :

$$q(j_1, j_2) * ..... * q(j_{n-1}, j_n)q(j_n, j_1) = q(j_n, j_{n-1}) * ..... * q(j_2, j_1)q(j_1, j_n) \quad (2.5)$$

dimostrazione :

Si suppone che la catena di Markov sia reversibile , quindi valgono le relazioni :

$$\pi(j_1)\ q(j_1, j_2) = \pi(j_2)\ q(j_2, j_1)$$

.

.

$$\pi(j_n)\ q(j_n, j_1) = \pi(j_1)\ q(j_1, j_n)$$

moltiplicandole fra loro si ottiene:

$$\pi(j_1) * ..... * \pi(j_n)q(j_1, j_2)q(j_2, j_3) * ..... * q(j_{n-1}, j_n)q(j_n, j_1) =$$

$$= \pi(j_1) * ..... * \pi(j_n)q(j_n, j_{n-1})q(j_{n-1}, j_{n-2}) * ..... * q(j_2, j_1)q(j_1, j_n)$$

che equivale alla (2.5) poichè per l'rridicibilità $\pi(j_1) * ..... * \pi(j_n) > 0$.
Viceversa si suppone che i tassi di transizione soddisfino la (2.5). Preso un qualsiasi stato di riferimento $j_0 \in \mathcal{X}$, esiste una sequenza di stati finita che collegano un generico stato j allo stato di riferimento (per l'irriducibilità della catena). Si pone :

$$\pi(j) = B\frac{q(j_0, j_1) * ..... * q(j_n, j)}{q(j, j_n) * ..... * q(j_1, j_0)} \tag{2.6}$$

dove a è una costante positiva e l'irriducibilità assicura che $\pi(j) > 0$. L'insieme $\{\pi(j) > 0 \;\; : \;\; j \in \mathcal{X}\}$ soddisfa la (2.1) e , giacchè la catena è stazionaria , la $\sum_{j\in\mathcal{X}}$ non può essere infinita . Quindi la costante B può esssere scelta in modo da rendere unitaria tale somma. Allora per il teorema (2.2) la catena è reversibile e $\pi(j)$ è la sua distribuzione all'equilibrio.

La (2.5) dice che la probabilità di percorrere un qualunque percorso in un verso euivale a quella di percorrere lo stesso cammino in verso opposto.

Si noti che la (2.6) è indipendente dalla particolare sequenza $j_1, ..., j_n$ che collega lo stato j_0 con lo stato j , questo poichè la (2.5) implica che :

$$\frac{q(j_1, j_n}{j_n, j_1} = \frac{q(j_1, j_2) * ....q(j_{n-1}, j_n)}{q(j_n, j_{n-1}) * ....q(j_2, j_1)}$$

cioè il rapporto precedente dipende solo dallo stato iniziale e dallo stato finale e non dal percorso che li collega.

I tassi di una catena di Markov irriducibile possono essre cambiati in diversi modi , pur mantenendo la reversibilità . Se per esempio , in una catena di Markov reversibile $q(j_1, j_2)$ viene modificato in $cq(j_1, j_2)$ e $q(j_2, j_1)$ in $cq(j_2, j_1)$, la catena rimane reversibile e mantiene la stessa distribuzione all'equilibrio (la verifica si ottiene immediatamente utilizzando la (2.1)) . Il seguente teorema estende il precedente risultato a un caso molto più generale.

TEOREMA (2.6) :

Sia x(t) una caetna di Markov irriducibile su $A \subset \mathcal{X}$ (spazio degli stati) e con dstribuzione all'equilibrio $\pi(j)\ j \in \mathcal{X}$. Si consideri la catena di Markov $\hat{x}(t)$ ottenuta ponendo :

$$\hat{q}(j_1, j_2) = cq(j_1, j_2) \quad \forall j_1 \in A \text{ e } \forall j_2 \in \mathcal{X} - A$$

$$\hat{q}(j_1, j_2) = q(j_1, j_2) \quad altrimenti$$

dove $c \geq 0$.
Allora $\hat{x}(t)$ rimane reversibile e ha distribuzione all'equilibrio

$$\hat{\pi}(j) = k\pi(j) \ \ se \ \ j \in A$$

$$\hat{\pi}(j) = kc\pi(j) \ \ se \ \ j \not\in A$$

dove k è la costante di normalizzazione.

Dimostrazione :
La dimostrazione si ottiene direttamente dal teorema (2.2). Infatti la distribuzione all'equilibrio $\hat{\pi}(j)$ di $\hat{x}(t)$ soddisfa l'quazione (2.1). La costante di normalizzazione è inoltre data da

$$k = \frac{1}{\sum_{j \in A} \pi(j) + \sum_{j \in \mathcal{X}} \pi(j)}$$

Osservazione : Se $c = 0$ (lemma del troncamento) la catena di Markov viene limitata allo spazio di stato $A \subset \mathcal{X}$, restando reversibile con distribuzione all'equilibrio :

$$\hat{\pi}(j) = \frac{\pi(j)}{\sum_{j \in A} \pi(j)} \ ; \ j \in A$$

Il lemma del troncamento viene spesso utilizzato per reti di code con stazioni a capacità limitata.

Si passa ora alla quasi-reversibilità e alla conseguente forma prodotto che consente un più facile calcolo della distribuzione all'equilibrio .

Capitolo 3

QUASI REVERSIBILITÀ

3.1 Definizione di quasi reversibilità

Quanto visto finora consente di verificare se un insieme di numeri $\pi(j)$, è la distribuzione all'equilibrio (invariante e limite) per un sistema modellabile tramite una catena di Markov a tempo continuo regolare irriducibile. Infatti data una tale catena se esiste $\pi(i) > 0 \; \forall i \in \mathcal{X}$ che soddisfa :

$$\sum_{i \in \mathcal{X}} \pi(i) = 1$$

$$\pi(i)\; q(i,j) = \pi(j)\; q(j,i) \quad \forall i,j \in \mathcal{X}$$

per il teorema (1.1) e per il teorema (1.2) , $\pi(j)$ è la distribuzione all'equilibrio , invariante e limite della catena di Markov. Per il teorema (2.2) la catena è anche reversibile.

Si pone ora il problema della determinazione dell'insieme $\pi(.)$ da studiare . Il concetto di quasi resersibilità risolve questo problema, permettendo di esprimere la probabilità all'equilibrio di una rete di code quasi reversibile come prodotto delle distribuzioni delle singole code.

Definizione : una coda si dice quasi reversibile quando il suo stato $x(t)$ è una catena di Markov stazionaria tale che , fissato $t_0 \in \Re$, $x(t_0)$ è indipendente :

- dai processi di arrivo degli utenti di classe c $\forall t \geq t_0$ e $\forall c = 1, ..., C$ (C = numero di classi presenti)

- dai processi di partenza degli utenti di classe c $\forall t \leq t_0$ e $\forall c = 1, ..., C$.

Si forniscono due teoremi che illustrano le propità della quasi reversibilità (le dimostrazioni si trovano in Kelly).

TEOREMA (3.1)

Se una coda è quasi reversibile allora :

- i tempi di interarrivo degli utenti di classe c $\forall c = 1, .., C$ formano processi di Poisson omogenei ed indipendenti

- i tempi di interpartenza degli utenti di classe c $\forall c = 1, .., C$ formano processi di Poisson omogenei ed indipendenti

TEOREMA (3.2)

Una coda è quasi reversibile se e solo se , $\forall c = 1, ..., C$, si ha

$$\pi(x) \sum_{x' \in S(c,x)} q(x, x') = \sum_{x' \in S(c,x)} \pi(x') q(x', x)$$

dove $S(c, x)$ è l'insieme degli stati in cui la coda contiene un utente in più di classe c rispetto allo stato x.

Osservazione : la coda M-M-1 con una sola classe di utenti e con disciplina FCFS (o LCFS o PS) è sia reversibile , sia quasi reversibile . La reversibilità di tale coda si è già praticamente vista nel primo capitolo, poichè , in questo caso, l'equazione

$$\pi(i) \sum_{j \in \mathcal{X}} q(i, j) = \sum_{j \in \mathcal{X}} \pi(j) q(j, i)$$

e l'equazione

$$\pi(i)\, q(i,j) = \pi(j)\, q(j,i)$$

coincidono.
La quasi reversibilità si ottiene osservando che nelle code M-M-1 gli arrivi dopo t_0 sono indipendenti da $x(t_0)$ e che, essendo reversibile, anche le partenze prima di t_0 sono indipendenti da $x(t_0)$.

Si illustra con un esempio il perché la quasi reversibilità garantisce una forma chiusa (forma prodotto) della distribuzione all'equilibrio.

3.2 Reti di code quasi reversibili con routing di Markov

Si definisce rete chiusa, multiclasse,quasi reversibile, con routing di Markov, una rete così caratterizzata :

1) la rete è composta da M code indicizzate con $1 \leq i \leq M$

2) la rete è chiusa (numero di utenti costante)

3) ogni coda ha spazio di stato $\mathcal{X}_i$ e tassi di transizione $q_i(.,.)$

4) ogni coda è quasi reversibile con tassi di arrivo per ogni classe pari a λ_i^c e tassi di servizio pari a μ_i^c (c=1,...,C) e ammette distribuzione invariante $\pi(.)$

5) con probabilità r_{ij}^{cd} un utente di classe c che lascia la coda i-esima viene mandato nella coda j-esima cambiando la propria classe in d ($\sum_{jd} r_{ij}^{cd} = 1$). Le decisioni che riguardano il routing sono tutte indipendenti fra loro e indipendenti dalla passata evoluzione del processo che descrive la rete.

Per giungere a definire la rate matrix della catena di Markov , che descrive l'evoluzione della rete abbiamo bisogno della seguente osservazione .

Ossservazione : si suppone che , all'istante t , si verifichi l'arrivo di un utente di classe d nella coda j-esima , quando lo stato di tale coda è x_i. A seguito di tale arrivo lo stato x_j assumerà un nuovo valore. Si calcola , di seguito , la probabilità di questo nuovo valore.

$$P[x_j(t) = y_j | x_j(t^-) = x_j, t \; istante \; di \; arrivo \; di \; un \; utente \; di \; classe \; d] =$$

$$\lim_{\epsilon \to 0^+} P[x_j(\epsilon) = y_j | x_j(0) = x_j \;,\; A_\epsilon^{jd} \geq 1] =$$

dove A_ϵ^{jd} è l'insieme degli arrivi di un utente di classe d nella coda j nell'intevallo temporale $(0, \epsilon)$

$$= \lim_{\epsilon \to 0^+} \frac{P[x_j(\epsilon) = y_j \;,\; x_j(0) = x_j \;,\; A_\epsilon^{jd} \geq 1]}{P[x_j(0) = x_j, A_\epsilon^{jd} \geq 1]} =$$

sfruttando la quasi reversibilità della coda j , che implica l'indipendenza tra gli arrivi futuri e lo stato attuale , si può scrivere :

$$= \lim_{\epsilon \to 0^+} \frac{P[x_j(\epsilon) = y_j \;,\; x_j(0) = x_j \;,\; A_\epsilon^{jd} \geq 1]}{P[x_j(0) = x_j] P[A_\epsilon^{jd} \geq 1]} =$$

$$= \lim_{\epsilon \to 0^+} \frac{P[x_j(\epsilon) = y_j \;,\; A_\epsilon^{jd} \geq 1 | x_j(0) = x_j]}{P[A_\epsilon^{jd} \geq 1]} =$$

supponendo che $x_j, y_j \in A^{jd}$ (insieme degli arrivi di un utente di classe j nella coda j)

$$= \lim_{\epsilon \to 0^+} \frac{\epsilon \; q(x_j, y_j) + \sigma(\epsilon)}{\epsilon \; \lambda_j^d + \sigma(\epsilon)]} =$$

dove $\sigma(\epsilon)$ è un infinitesimo, per ϵ tendente a zero di ordine superiore al primo

$$= \frac{q(x_j, y_j)}{\lambda_j^d}.$$

Quindi, dati due vettori di stato $\underline{x} = (x_1, ..., x_M)$ e $\underline{y} = (y_1, ..., y_M)$, i tassi di transizione della catena di Markov che descrive l'evoluzione della rete sono :

$$q(\underline{x}, \underline{y}) = \frac{q_i(x_i, y_i) \; q_j(x_j, y_j)}{\lambda_j^d} r_{ij}^{cd} \tag{3.1}$$

$$(\underline{x}, \underline{y}) \in \mathcal{X} = \{(x_1, ..., x_M) : \textit{ la somma degli utenti è costante}\}$$

$$(\underline{x}, \underline{y}) \in T_{ij}^{cd} \ , \ i, j = 1, .., M \ , \ c, d = 1, .., C$$

dove T_{ij}^{cd} è l'insieme delle transizioni ($\underline{x}$,$\underline{y}$) che corrispondono ad utente di classe c che si sposta dalla coda i alla j , diventando di classe d. Si escludono per semplictà transizioni interne ad una coda (cambiamenti di classe al suo interno).

Per giustificare la (3.1) si può affermare che $q(\underline{x}, \underline{y})$ è uguale al tasso di partenza di un utente di classe c dalla coda i-esima , per la probabilità che questo utente vada nella coda j-esima cambiando la sua classe in d, per la probabilità che ad un arrivo di classe d nella coda j corrisponda una transizione da x_j a y_j .

Vogliamo ora verificare che la rete di code data è modellabile tramite una catena di Markov, $x(t)$, reversibile e con distribuzione all'eaquilibrio data da:

$$\pi(\underline{x}) = K\pi_1(x_1) * ..... * \pi_M(x_M) \tag{3.2}$$

dove K è la costante di normalizzazione e le $\pi_i(x_i)$ sono le distribuzioni all' equilibrio delle code come se fossero isolate.

Per semplicità di notazione, la verifica della (3.2) verrà effettuata nel caso sia presente una sola classe di utenti , e le code siano tutte del tipo M-M-1 (o FCFS,o LCFS,o PS).

Sia $\mathcal{X} = \{(x_1, ..., x_M) : \sum_{k=1}^{M} = N\}$ lo spazio degli stati della catena di Markov $x(t)$, e siano $\underline{x}, \underline{y}$ appartenenti a $\mathcal{X}$. La (2.1) del teorema (2.2) implica :

$$\pi(\underline{x}) \ \frac{q_i(x_i, x_i - 1) \ q_j(x_j, x_j + 1)}{\lambda_j} \ r_{ij} = \pi(\underline{y}) \ \frac{q_i(x_i - 1, x_i) \ q_j(x_j + 1, x_j)}{\lambda_i} \ r_{ji}$$

sostituendo la (3.2) nella precedente e semplificando si ottiene

$$\pi_i(x_i) \ \pi_j(x_j) \ \frac{q_i(x_i, x_i - 1) \ q_j(x_j, x_j + 1)}{\lambda_j} \ r_{ij} =$$

$$= \pi_i(x_i - 1) \ \pi_j(x_j + 1) \ \frac{q_i(x_i - 1, x_i) \ q_j(x_j + 1, x_j)}{\lambda_i} \ r_{ji}$$

ricordando che la distribuzione all'equilibrio di una coda M-M-1 è

$$\pi(x) = \pi(0)\Big(\frac{\lambda}{\mu}\Big)^x$$

e sostituendo si ottiene

$$\Big(\frac{\lambda_i}{\mu_i}\Big)^{x_i}\Big(\frac{\lambda_j}{\mu_j}\Big)^{x_j}\mu_i r_{ij} = \Big(\frac{\lambda_i}{\mu_i}\Big)^{x_i-1}\Big(\frac{\lambda_j}{\mu_j}\Big)^{x_j+1}\mu_j r_{ji}$$

che si riduce alla

$$\lambda_i \; r_{ij} = \lambda_j \; r_{ji} \tag{3.3}$$

Se esistono $\lambda_1, ..., \lambda_M$ tali da soddisfare la (3.3), per ogni transizione di un utente dalla coda i-esima alla coda j-esima $\forall j, i = 1, ..., M$, per i teoremi (2.1) (1.1) (1.2) la rete è reversibile e

$$\pi(\underline{x}) = K\pi_1(x_1) * ..... * \pi_M(x_M) \tag{3.4}$$

è la sua disribuzione all'equilibrio,invariante e limite. Infatti la catena di Markov è irriducibile su $\mathcal{X} = \{(x_1, ...x_M) : \sum_{k=1}^{M} = N\}$, regolare, $\pi(\underline{x}) > 0$, e esiste k tale che

$$\sum_{\underline{x}\in\mathcal{X}} \pi(\underline{x}) = 1$$

poichè l'insieme degli stati è finito e quindi la somma precedente non può evidentemente divergere .

Se la (3.3) non può essere verificata , la rete di code non può essere modellata tramite una catena di Markov reversibile con distribuzione all'equilibrio

$$\pi(\underline{x}) = K\pi_1(x_1) * ..... * \pi_M(x_M)$$

perché il teorema (2.2) è condizione necessaria e sufficiente per la reversibilità .

Quanto visto in questo paragrafo è la dimostrazione del seguente teorema.

TEOREMA (3.3) :

Una rete di code , chiusa ,monoutente, con routing di Markov , formata da M code $(i = 1, ..., M)$ M-M-1 (discipline FCFS o LCFS o PS), con tassi di arrivo λ_i e tassi di servizio μ_i , è modellabile con una catena di Markov , irriducibile su $\mathcal{X} = \{(x_1, ...x_M) : \sum_{k=1}^{M} = N\}$, reversibile , e con distribuzione all'equilibrio

$$\pi(\underline{x}) = K\pi_1(x_1) * ..... * \pi_M(x_M) \; ; \; \underline{x} \in \mathcal{X}$$

se e solo se esiste un insieme di numeri positivi $\{\lambda_i\}$ tale che per ogni transizione risulti:

$$\lambda_i r_{ij} = \lambda_j r_{ji} \;\; ; \;\; \forall i, j \in \mathcal{X}$$

dove i λ_i saranno proprio i tassi di arrivo delle code M-M-1, e dove $\pi_i(x_i)$ sono le distribuzione all'equilibrio di tali code come se fossero isolate.

Osservazione : con la (3.3) si pretende la reversibilità . Se tale equazione non è verificata non è assolutamente detto che non esista una catena di Markov stazionaria che descrive la rete precedentemente esposta. Il problema poteva essere affrontato imponendo

$$1) \; \pi(\underline{x}) \sum_{\underline{y} \in \mathcal{X}} q(\underline{x}, \underline{y}) = \pi(\underline{y}) \sum_{\underline{y} \in \mathcal{X}} q(\underline{y}, \underline{x})$$

invece di

$$2) \; \pi(\underline{x}) \; q(\underline{x}, \underline{y}) = \pi(\underline{y}) \; q(\underline{y}, \underline{x})$$

Nel caso 1) si sarebbero ottenuto un sistema lineare omogeneo di M equazioni (M-1 linearmente indipendenti) con M incognite $\lambda_1, ..., \lambda_m$, quindi sempre risolubile (per il teorema di Rouchè - Capelli). Il sistema che deriva dalla condizione 2) non sempre è risolubile in quanto il numero di equazioni può essere superiore al numero di incognite.

Si è scelto di imporre la reversibilità , perché in questa sede ci si occupa di reti di code con struttura ad albero. In questo caso il sistema di equazioni (3.3) (lineare ed omogeneo) presenta M-1 equazioni (numero di rami in un albero con M nodi) , M incognite ed è quindi sempre risolubile.

Da quanto detto prima si ha il seguente teorema.

TEOREMA (3.4) :

Una rete di code, chiusa, monoutente, con routing di Markov, e struttura ad albero, formata da M code ($i = 1, ..., M$) M-M-1 (discipline FCFS o LCFS o

PS), con tassi di arrivo λ_i e tassi di servizio μ_i, è modellabile con una catena di Markov , irriducibile su $\mathcal{X} = \{(x_1, ...x_M) : \sum_{k=1}^{M} = N\}$, reversibile, e con distribuzione all'equilibrio (unica per il teorema (1.2))

$$\pi(\underline{x}) = K\pi_1(x_1) * ..... * \pi_M(x_M) \; ; \; \underline{x} \in \mathcal{X}$$

dove $\pi_i(x_i)$ sono le distribuzione all'equilibrio delle code M-M-1 come se fossero isolate.

Anologamente a quanto visto per le reti chiuse si può dimostrare il teorema che segue. Il teorema seguente è valido per reti di code aperte aventi una sola coda comunicante bidirezionalmente con l'esterno (vedere paragrafo (4.3).

TEOREMA (3.5) :

Una rete di code, aperta, monoutente, con routing di Markov, e struttura ad albero, formata da M code ($i = 1, ..., M$) M-M-1 (discipline FCFS o LCFS o PS), con tassi di arrivo λ_i e tassi di servizio μ_i , è modellabile con una catena di Markov, irriducibile su $\mathcal{X} = \{(x_1, ...x_M) : \sum_{k=1}^{M} = N\}$, reversibile, e con distribuzione all'equilibrio (unica per il teorema (1.2))

$$\pi(\underline{x}) = K\pi_1(x_1) * ..... * \pi_M(x_M) \; ; \; \underline{x} \in \mathcal{X}$$

dove $\pi_i(x_i)$ sono le distribuzione all'equilibrio delle code M-M-1 come se fossero isolate.

L'unica differenza sostanziale, per le reti aperte, è che il sistema di equazioni che si ottiene imponendo :

$$\pi(\underline{x}) = K\pi_1(x_1) * ..... * \pi_M(x_M) \; ; \; \underline{x} \in \mathcal{X}$$

non è più omogeneo , ed ammette sempre un unica soluzione (per reti aventi struttura ad albero con collegamenti bidirezionale, con una sola coda comunicante con l'esterno).

Nel prossimo capitolo si estenderanno tali risultati a reti di code a capacità limitata e con tassi di servizio dipendenti dallo stato.

Capitolo 4

RETI A CAPACITÀ LIMITATA

4.1 Reti chiuse a capacità limitata

In questo capitolo si estenderanno i risultati del paragrafo (3.2) a reti di code le cui stazioni sono a capacità limitata e con tassi di servizio dipendenti dal numero di utenti presenti. A tal proposito si considerino M code modellabili con catene di Markov $x_i(t)$ regolari ,aventi spazio di stato

$$\mathcal{X}_i = \{0, 1, ..., N_i\}\ ; \quad i = 1, ..., M$$

e tassi di transizione

$$q(n_i, m_i) = \lambda_i\ 1(n_i < N_i) \quad per\ m_i = T_{.i}(n_i) \quad n_i, m_i \in \mathcal{X}_i$$

$$q(n_i, m_i) = \mu_i(n_i) \quad per\ m_i = T_{i.}(n_i) \quad n_i, m_i \in \mathcal{X}_i$$

dove n_i, m_i rappresentano il numero di utenti in coda , N_i la capacità della coda , $T_{.i}(n_i)$ la nascita di utente nella coda i-esima ,$T_{i.}(n_i)$ la morte di utente . I tassi di transizione devono soddisfare le seguenti condizioni :

$$\lambda_i > 0\ \ \forall i = 1, ..., M\ (dimensione\ di\ frequenza)$$

$$\mu_i(0) = 0\ e\ \mu_i(n_i) > 0\ \forall i = 1, ..., M\ , \forall n_i = 1, .., N_i$$

Tali condizioni servono a garantire l'irriducibilità su $\mathcal{X}_i$ della catena di Markov $x_i(t)$. La coda $x_i(t)$ ha quindi il seguente comportamento : rimane nello stato n_i

per un tempo τ distribuito esponenzialmente con parametro $\beta(n_i) = \lambda_i \; 1(n_i < N_i) + \mu_i(n_i)$, intercorso il quale, passa nello stato $n_i + 1$ con probabilità

$$\Gamma_{n_i+1} = \frac{\lambda_i \; 1(n_i < N_i)}{\lambda_i \; 1(n_i < N_i) + \mu_i(n_i)}$$

e in $n_i - 1$ con probabilità

$$\Gamma_{n_i-1} = \frac{\mu_i(n_i)}{\lambda_i \; 1(n_i < N_i) + \mu_i(n_i)}$$

Dopo la transizione l'evoluzione ricomincia indipendentemente da quanto successo prima.

Si vuole determinare ora la distribuzione all'equilibrio della catena di Markov $x_i(t)$. Dalla (2.1) si ricava $(n_i > 0)$:

$$\pi_i(n_i) \; q(n_i, n_i - 1) = \pi_i(n_i - 1) \; q(n_i - 1, n_i)$$

da cui

$$\pi_i(n_i) \; \mu_i(n_i) = \pi_i(n_i - 1) \; \lambda_i \; 1(n_i - 1 < N_i)$$

quindi

$$\pi_i(n_i) = \pi_i(0) \prod_{r=1}^{n_i} \frac{\lambda_i \; 1(n_i < N_i)}{\mu_i(r)} \tag{4.1}$$

con

$$\pi_i(0) = \frac{1}{\sum_{n=1}^{N_i} \prod_{r=1}^{n_i} \frac{\lambda_i}{\mu_i(r)}}$$

Essendo $x_i(t)$ una catena di Markov regolare irridicibile su $\mathcal{X}_i$, la (4.1) è la distribuzione all'equilibrio,invariante e limite. Inoltre, per il teorema (2.2) $x_i(t)$ è anche reversibile.

Si connettono ora le M code in modo che la struttura sia ad albero. Supponiamo che la rete risultante soddisfi le seguenti specifiche:

1) il routing è Markov

2) la rete è chiusa (con numero di utenti N)

3) quando un utente richiede servizio ad una stazione satura viene bloccato e

rimane nella stazione in cui si trovava (questo tipo di bloccaggio prende il nome di 'block and hold')

4) la strutura della rete è ad albero con collegamenti bidirezionali fra code adiacenti

La catena di Markov $x(t)$ che descrive l'evoluzione della rete ha spazio di stato :

$$\mathcal{X} = \{(n_1, ..., n_M) : n_i \in \mathcal{X}_i \ , \ \sum_{i=1}^{M} = N\}$$

Per il calcolo dei tassi di transizione della rete si osserva che la probabilità con la quale l'arrivo di un utente, all'istante t, provoca la transizione $n_j \rightarrow n_{j+1}$, è :

$$P[x_j(t) = n_{j+1} | x_j(t^-) = n_j \ , t \ istante \ arrivo \ utente \ ] = 1(n_j < N_j) \qquad (4.2)$$

cioè tale probabilità è unitaria se la j-esima coda non è satura, mentre è uguale a zero se la coda j-esima presenta già un numero di utenti pari alla sua capacità . Per calcolare la (4.2) si può utilizzare il formalismo descritto nel paragrafo (3.2)

I tassi di transizione di $x(t)$ sono quindi

$$q(\underline{n}, \underline{m}) = \mu_i(n_i) \ r_{ij} \ 1(n_j < N_j) \quad ; \quad \underline{n}, \underline{m} \in \mathcal{X} \ , \ \underline{m} = T_{ij}(\underline{n}) \qquad (4.3)$$

dove $T_{ij}(\underline{n}) = (n_1, .., n_{n_i - 1}, .., n_{j+1}, .., n_M)$ con $\underline{n} = (n_1, .., n_i, .., n_j, .., n_M) \in \mathcal{X}$ e r_{ij} è la probabilità che utente lasciante la coda i-esima venga indirizzato verso la coda j-esima ($r_{ij} = 0$ se le code i, j non sono collegate) .

Si verifica ora che $x(t)$ ha distribuzione all'equilibrio

$$\pi(\underline{n}) = \left[G_N \prod_{i=1}^{M} \pi_i(n_i) \right] \ 1(\underline{n} \in \mathcal{X}) \qquad (4.4)$$

con

$$\pi_i(n_i) = \pi_i(0) \prod_{r=1}^{n_i} \frac{\lambda_i \ 1(n_i < N_i)}{\mu_i(r)}$$

I λ_i della (4.4) verranno determinati (non univocamente) risolvendo un sistema lineare omogeneo.

Dalla (2.1) si ha :

$$\pi(\underline{n}) \ q(\underline{n}, \underline{m}) = \pi(\underline{m}) \ q(\underline{m}, \underline{n}) \ ; \quad \underline{n}, \underline{m} \in \mathcal{X} \qquad (4.5)$$

se $\underline{n}, \underline{m} \notin \mathcal{X}$, o se i, j non sono collegate l'equazione (4.5) è banalmente verificata . Altrimenti ponendo $\underline{n} = (n_1, .., n_i, .., n_j, .., n_M)$ e $\underline{m} = (n_1, .., n_i - 1, .., n_j + 1, .., n_M)$ si ottiene

$$\pi(\underline{n})\ \mu_i(n_i)\ r_{ij}\ 1(n_j < N_j) = \pi(\underline{m})\ \mu_j(n_j + 1)\ r_{ji}\ 1(n_i - 1 < N_i)$$

dove si è semplicemente sostituita la (4.3) nella (4.5).

Utilizzando la (4.4) e (4.1)

$$\pi(\underline{n})\ \mu_i(n_i)\ r_{ij}\ 1(n_j < N_j) =$$
$$= \pi(\underline{n})\ \frac{\mu_i(n_i)}{\lambda_i\ 1(n_i \leq N_i)} \frac{\lambda_j\ 1(n_j + 1 \leq N_j)}{\mu_j(n_{j+1})}\ \mu_j(n_j + 1)\ r_{ji}\ 1(n_i - 1 < N_i)$$

dove si è moltiplicato e diviso per quantità strettamente maggiori di zero ($\underline{n}, \underline{m} = T_{ij}(\underline{n}) \in \mathcal{X}$). Sotto queste ipotesi tutte le funzioni $1(.)$ presenti ,nella precedente equazione assumono valore unitario. Semplificando si ottiene

$$\lambda_i\ r_{ij} = \lambda_j\ r_{ji} \tag{4.6}$$

L'equazione (4.6), poichè la rete ha struttura ad albero con rami bidirezionali, formano un sistema di $M - 1$ equazioni lineari omogenee, in M incogite $(\lambda_1, ..., \lambda_M)$. Tale sistema ha infinite soluzioni formanti uno spazio vettoriale di dimensione unitaria, si può quindi fissare a piacere un incognita e determinare le altre di conseguenza.

Poichè il sistema (4.6) è sempre risolubile e la catena di Markov $x(t)$ è regolare e irriducibile su $\mathcal{X} = \{(n_1, ..., n_M) : n_i \in \mathcal{X}_i, \sum_{i=1}^{M} n_i = N\}$, la sua distribuzione all'equilibrio (unica) è :

$$\pi(\underline{n}) = G_N^{-1} \prod_{i=1}^{M} \prod_{r=1}^{n_i} \frac{\lambda_i}{\mu_i(r)} \quad ; \quad \underline{n} \in \mathcal{X} \tag{4.7}$$

Inoltre la rete è reversibile. La costante di normalizzazione G_N^{-1} vale

$$G_N^{-1} = \frac{1}{\sum_{\underline{n} \in \mathcal{X}} \prod_{i=1}^{M} \prod_{r=1}^{n_i} \frac{\lambda_i}{\mu_i(r)}} \tag{4.8}$$

Osservazione : la (4.7) è unica anche se i λ_i non sono univocamente determinati. La verifica di questo si vedrà in un caso particolare nel prossimo paragrafo.

Osservazione : la costante di normalizzazione non può essere calcolata, a livello pratico, tramite la (4.8) poichè l'insieme $\mathcal{X}$ contiene un numero di stati troppo grande (del ordine dei miliardi anche per modelli di piccola dimensione).

4.2 Un caso particolare di reti chiuse con struttura ad albero

Si analizza, in questo paragrafo un caso particolare di reti chiuse con struttura ad albero : le reti di code cosiddette a stella multipla. Tali reti possono essere utilizzate per modellare un sistema FMS (flexible manufactering system). Per il modello illustrato di seguito è stato progettato un codice eseguibile su Personal Computer riportato in appendice.

Consideriamo una rete di di code costituita da una coda centrale (trasporto principale) collegata a M code (trasporti locali). A sua volta il trasporto locale i-esimo, con $i = 1, ..., M$,è collegato a F^i code (stazioni serventi). I collegamenti sono tutti bidirezionali e, per quanto dettto, la rete è chiusa con struttura ad albero. Si è soliti suddividere le code della rete in celle. Una cella, indicizzata con $h = 0$ è formata solamente dal trasporto principale, le altre M celle sono formate da un trasporto locale e dalle sue stazioni serventi. In totale ci sono quindi $M+1$ celle ciascuna formata da F^j+1 nodi ($F^0 = 0$). Per quanto riguarda la notazione si indicherà ,per esempio ,con n_i^h il numero di utenti nella coda i ($i = 0, ..., F^h$) della cella h (h=0,.....,M). Se $i = 0$ e $h \neq 0$ si farà riferimemento al trasporto locale della h-esima cella ,se $h = 0$ (necessariamente i=0) si farà riferimento al trasporto principale. La precedente notazione verrà utilizzata per tutti i simboli presi in considerazione.

Continuando nella desrizione del modello, ogni coda è descritta da una catena di Markov $x_i^h(t)$ avente spazio degli stati

$$\mathcal{X}_i^h = \{0, 1, ..., N_i^h\} \; ; \quad h = 0, ..., M \; , \; i = 1, ..., F^h$$

con tassi di transizione

$$q(n_i^h, m_i^h) = \lambda_i^h \; 1(n_i^h < N_i^h) \qquad per \; m_i^h = T_{.i}(n_i^h) \quad n_i^h, m_i^h \in \mathcal{X}_i^h$$

$$q(n_i^h, m_i^h) = \mu_i^h(n_i^h) \qquad per \; m_i^h = T_{i.}(n_i^h) \quad n_i^h, m_i^h \in \mathcal{X}_i^h$$

dove si pretende che $\mu_i^h(0) = 0$ e $\mu_i^h(n_i^h) > 0$ per ogni valore assunto da i, h, n_i^h. Per la connessione si fanno le seguenti ipotesi :

1) il routing è Markov con probabiltà p_0^h che un utente, che lascia il trasporto principale, venga indirizzato verso il trasporto locale della h-esima cella ($h = 1, .., M$), mentre p_i^h rappresenta la probabilità che un utente, che lascia il trasporto della h-esima cella, venga indirizzato verso la coda i della stessa cella ($i = 1, .., F^h$).

2) il bloccaggio è di tipo 'block and hold'

3) la rete è chiusa, con N utenti, ed ha struttura ad albero a collegamenti bidirezionali

Lo spazio di stato della catena di Markov $x(t)$ che descrive l'evoluzione delle rete è

$$\mathcal{X} = \{(n_0^0, .., n_{F^M}^M) \; ; \; n_i^h \in \mathcal{X}_i^h \; , \; h = 0, .., M \; , \; i = 0, .., F^h \; , \; \sum_{h=0}^{M} \sum_{i=0}^{F^h} n_i^h = N\}$$

Procedendo come nel paragrafo (4.1) si ottengono i tassi di transizione di $x(t)$:

$$q(\underline{n}, \underline{m}) = \mu_0^0(n_0^0) \; p_0^h \; 1(n_0^h < N_0^h) \;\; ; \;\; \underline{m} = T^{0h}(\underline{n})$$

$$q(\underline{n}, \underline{m}) = \mu_0^h(n_0^h) \; (1 - \sum_{i=1}^{F^h} p_i^h) \; 1(n_0^0 < N_0^0) \;\; ; \;\; \underline{m} = T^{h0}(\underline{n})$$

$$q(\underline{n}, \underline{m}) = \mu_0^h(n_0^h) \; p_c^h \; 1(n_c^h < N_c^h) \;\; ; \;\; \underline{m} = T_{0c}^h(\underline{n})$$

$$q(\underline{n}, \underline{m}) = \mu_c^h(n_c^h) \; 1(n_0^h < N_0^h) \;\; ; \;\; \underline{m} = T_{c0}^h(\underline{n})$$

dove

$\underline{n}, \underline{m} \in \mathcal{X}$, $h = 1, .., M$, $c = 1, .., F^h$

$T^{0h}(.)$: indica la transizione di un utente dal trasporto principale al trasporto locale h-esima cella.

$T^{h0}(.)$: indica la transizione inversa di $T^{0h}(.)$

$T^h_{0c}(.)$: indica la transizione di un utente dal trasporto locale della cella h alla stazione c della stessa cella

$T^h_{c0}(.)$: indica la transizione inversa di $T^h_{0c}(.)$

Inoltre $1 - \sum_{i=1}^{F^h} p^h_i$ è la probabilità che un, lasciante il trasporto locale h, venga indirizzato al trasporto principale.

La rete descritta è un caso particolare delle reti illustrate nel paragrafo (4.1). Infatti tutte le ipotesi coincidono, e in questo caso, 'l'albero di connessione' ha tre livelli (nel paragrafo precedente poteva essere generico). Quindi valgono tutti i risultati precedentemente ottenuti.

In particolare, la catena di markov $x(t)$ è reversibile e ha distribuzione all'equilibrio, invariante e limite :

$$\pi(\underline{n}) = G_N^{-1} \prod_{h=0}^{M} \prod_{i=0}^{F^h} \prod_{r=1}^{n^h_i} \frac{\lambda^h_i}{\mu^h_i(r)} \quad ; \quad \underline{n} \in \mathcal{X} \tag{4.9}$$

dove i λ^h_i sono determinati (non univocamente) dalle equazioni

$$\lambda^0_0 \, p^h_0 = \lambda^h_0 \, (1 - \sum_{i=1}^{F^h} p^h_i) \quad ; \quad h = 1, .., M \tag{4.10}$$

$$\lambda^h_c = \lambda^h_0 \, p^h_c \quad ; \quad h = 1, .., M \ , \ c = 1, .., F^h \tag{4.11}$$

Osservazione : la (4.10) è la (4.6) scritta per le transizioni $T^{0h}(.)$ e $T^{h0}(.)$, mentre la (4.11) è sempre (4.6) scritta per le transizioni $T^h_{0c}(.)$ e $T^h_{c0}(.)$.

Il sistema formato dalle (4.10) e dalle (4.11) è sempre risolubile , essendo composto da $F^1 + .. + F^M + M$ equazioni lineari omogenee aventi $F^1 + .. + F^M + M + 1$ incognite (λ^h_i).

Osservazione : la soluzione generale del sistema formato dalle (4.10) (4.11) (spazio vettoriale di dimensione uno), pur non essendo unica, non comporta la non unicità della (4.9). Infatti fissato, per semplicità , $\lambda^h_0 = K$ si ha che :

$$\lambda^h_0 = K \frac{p^h_0}{1 - \sum_{i=1}^{F^h} p^h_i} = \lambda^h_0 \, p^h_0 \, b^h$$

$$\lambda_0^h = \lambda_0^h \; p_c^h$$

sostituendo nella (4.9)

$$\pi(\underline{n}) = G_N K^N \left(\prod_{h=1}^{M} p_0^h b^h \right)^{N-n_0^0} \prod_{h=1}^{M} \prod_{i=1}^{F^h} (p_i^h)^{n_i^h} \prod_{h=0}^{M} \prod_{i=0}^{F^h} \prod_{r=1}^{n_i^h} \frac{1}{\mu_i^h(r)} \tag{4.12}$$

dove $b^h = 1 - \sum_{i=1}^{F^h} p_i^h$.

La (4.12) mette in luce che $\lambda_0^0 = K$ può essere conglobato nella costante di normalizzazione. Ciò implica che è possibile fissare λ_0^0 (o qualsiasi altra variabile) in maniera da determinare l'ordine di grandezza della costante di normalizzazione. Questo può risultare utile nel caso in cui la medesima diventi o troppo grande o troppo piccola.

4.3 Reti aperte con struttura ad albero

Il modello è quello del paragrafo (4.1), tranne per il fatto che gli utenti possono arrivare dall'esterno secondo un processo di Poisson di parametro γ, efinire nella coda i con probabilità r_{0i}, e possono uscirne con probabilità $r_{i0} = 1 - \sum_j r_{ij}$. Lo spazio di stato della catena di Markov che modella la rete è

$$\mathcal{X} = \mathcal{X}_1 x....x \mathcal{X}_M \; ; \; \mathcal{X}_i = \{0, ..., N_i\}$$

e i tassi di transizione

$$q(\underline{n}, \underline{m}) = \gamma \; r_{0i} \; 1(n_i < N_i) \;\; ; \quad \underline{m} = T_{.i}(\underline{n})$$

$$q(\underline{n}, \underline{m}) = \mu_i(n_i) \; r_{0i} \;\; ; \quad \underline{m} = T_{i.}(\underline{n})$$

$$q(\underline{n}, \underline{m}) = \mu_i(n_i) \; r_{ij} \; 1(n_j < N_j) \;\; ; \quad \underline{m} = T_{ij}(\underline{n})$$

con $\underline{n}, \underline{m} \in \mathcal{X}$.

Supponendo che la distribuzione all'equilibrio di $x(t)$ sia in forma di prodotto ,imponendo la (2.2) ,per la transizione $T_{.i}(.)$ e $T_{i.}(.)$ si ottiene :

$$\lambda_i \; r_{i0} = \gamma \; r_{0i} \tag{4.13}$$

per le transizioni $T_{ij}()$

$$\lambda_i \; r_{ij} = \lambda_j \; r_{ji} \tag{4.14}$$

Osservazione : la (4.13) mette in luce che se degli utenti possono entrare in una coda, gli stessi utenti devono poterne anche uscire.
In ogni caso, per reti aperte aventi struttura ad albero, non sempre le (4.13) e (4.14) possono essere verificate. Considerando infatti due code collegate fra loro, che possono entrambe scambiare con l'esterno, le (4.13) determinano già λ_1 e λ_2 e quindi la (4.14), in generale non può essere verificata. In base all'osservazione precedente si aggiunge l'ipotesi che una sola delle M code possa comunicare con l'esterno.

In questo caso le (4.13) (4.14) sono verificate ed hanno soluzione unica ,quindi $x(t)$ è reversibile ed ha distribuione all'equilibrio

$$\pi(\underline{n}) = \prod_{i=1}^{M} \pi_i(n_i) \quad ; \quad \underline{n} \in \mathcal{X} \tag{4.15}$$

con

$$\pi_i(n_i) = \pi_i(0) \prod_{r=1}^{n_i} \frac{\lambda_i}{\mu_i(r)} \quad ; \quad n_i \in \mathcal{X}_i$$

La (4.15) rappresenta anche la distribuzione invariante e limite.

Capitolo 5

PROCESSI MIGRATORI

In questo capitolo si giungerà alla reversibilità e alla forma prodotto del modello del paragrafo (4.2), mediante la teoria dei processi migratori e, in particolare, mediante un metodo descritto da Pollet in [13]. Intuitivamente si possono considerare i processi migratori come catene di Markov che descrivono gli spostamenti (migrazioni) di individui (utenti) tra più colonie (stazioni). Le singole colonie sono processi nascita-morte.

5.1 Processi nascita e morte

Si consideri una catena di Markov $x_i(t)$ con spazio di stato

$$\mathcal{X}_i = \{0, 1, ..., N_i\}$$

dove N_i può tendere all'infinito .
I tassi di transizione sono :

$$q(n_i, m_i) = \psi(n_i) \ \ ; \quad se\ m_i = T_{.i}(n_i)$$

$$q(n_i, m_i) = \phi(n_i) \ \ ; \quad se\ m_i = T_{i.}(n_i)$$

dove $n_i, m_i \in \mathcal{X}_i$ e

- $m_i = T_{.i}(n_i)$: indica la nascita di un utente, cioè $m_i = n_i + 1$

- $m_i = T_{i.}(n_i)$: indica la morte di un utente, cioè $m_i = n_i - 1$

Inoltre le funzioni $\psi(.)$ $\phi(.)$ devono soddisfare le condizioni :

- $\psi(n_i) > 0$ con $0 \leq n_i < N_i$

- $\psi(n_i) = 0$ con $n_i = N_i$

- $\phi(n_i) > 0$ con $0 < n_i \leq N_i$

- $\phi(n_i) = 0$ con $n_i = 0$

Le condizioni 1) e 3) servono a garantire l'irriducibiltà di $x_i(t)$ in $\mathcal{X}_i$, la 2) impedisce la nascita di un utente quando la colonia è satura, e la 4) impedisce la morte di utente quando non sono presenti utenti nella colonia.

Una catena di Markov di questo tipo viene chiamata processo di nascita-morte in quanto fissato un generico stato n_i sono possibili solo le seguenti transizioni

- $n_i \rightarrow n_i + 1$: nascita di un utente

- $n_i \rightarrow n_i - 1$: morte di un utente

Nel capitolo 4) si era già visto un caso particolare di processo nascita morte in cui $\psi(n_i) = \lambda_i \; 1(n_i < N_i)$.

TEOREMA (5.1)

Un processo di nascita e di morte è reversibile e ha distribuzione all'equilibrio (invariante e limite) :

$$\pi(n) = \pi(0) \prod_{r=1}^{n} \frac{\psi(r-1)}{\phi(r)} \quad ; \quad n \in \mathcal{X} = \{0, ..., N\} \tag{5.1}$$

con $\mathcal{X}$ spazio degli stati.

dimostrazione

Dato che un processo di nascita e morte è una catena di Markov si possono applicare i risultati visti per queste ultime.
Dalla (2.1) si ottiene

$$\pi(n)\psi(n) = \pi(n+1)\phi(n+1)$$

da cui

$$\pi(1) = \pi(0)\frac{\psi(0)}{\phi(1)}$$

$$\pi(2) = \pi(2)\frac{\psi(1)}{\phi(2)} = \pi(0)\frac{\psi(0)}{\phi(1)}\frac{\psi(1)}{\phi(2)}$$

.

.

$$\pi(n) = \pi(0)\prod_{r=1}^{n}\frac{\psi(r-1)}{\phi(r)}$$

Quindi per il teorema (2.2), (1.1) e (1.2) il processo di nascita e morte è reversibile e ha distribuzione all'equilibrio, invariante e limite data dalla (5.1).

5.2 Processi Migratori

I processi migratori si ottengono connettendo più processi nascita e morte. In generale non è assolutamente detto che la connessione di processi nascite e morete (reversibili) sia ancora un processo reversibile. Infatti nel capitolo 4) si sono dovute fare varie ipotesi (routing di Markov, struttura ad albero con rami bidirezionali, ecc..) affinchè la rete risultante sia reversibile. Per esempio i collegamenti bidirezionali sono condizione necessaria per la reversibilità della rete. Nel prossimo paragrafo si esprrà un metodo, descritto da Pollet in [13], per collegare tra loro un insieme i processi reversibili, in modo tale che il processo risultante sia ancora reversibile e con distribuzione all'equilibrio prodotto delle distribuzioni all'equilibrio dei singoli processi come se fossero isolati.

Osservazione : il formalismo di Pollet non riguarda solamente la connessione di processi nascita-morte, ma più in gnerale la connessione di processi di Markov reversibili.

5.3 Formalismo di Pollet

Supponiamo di voler connettere tra loro M catene di Markov reversibili dove :

$\mathcal{X}_i$ è lo spazio degli stati della i-esima catena

Q_i è la matrice di transizione della i-esima catena di Markov

Sia $\mathcal{X}_i^{'} \subset \mathcal{X}_i \; x \; \mathcal{X}_i$ un insieme di coppie di stati, corrispondenti a possibili transizioni. Sia inoltre $\{A_i(c),\ c = 1, ..., C_i\} \subseteq \mathcal{X}_i^{'}$ soddisfacente alle proprietà :

- $A_i(.) \cap A_j(.) = \emptyset \; \forall i \neq j$

- se $(x, y) \in A_i(c)$ per qualche $c = 1, .., C_i$, allora $(y, x) \not\in U_i$

dove $U_i = \bigcup_{c=1}^{C_i} A_i(c)$.
Il processo dopo la connessione ha spazio degli stati :

$$\mathcal{X} = \mathcal{X}_1 \; x .....x \; \mathcal{X}_M$$

e i tassi di transizione sono definiti come segue

1) Sia $\underline{x} = (x_1, .., x_j^{'}, .., x_k, .., x_M) \in \mathcal{X}$ e $\underline{x}^{'} = (x_1, .., x_j, .., x_k^{'}, .., x_M) \in \mathcal{X}$, se $(x_j, x_j^{'}) \in A_j(c)$ $c = 1, .., C_j$ e se $(x_k, x_k^{'}) \in A_k(d)$ $d = 1, .., C_k$, allora

$$q(\underline{x}, \underline{x}^{'}) = q_j(x_j^{'}, x_j) q_k(x_k, x_k^{'}) \gamma_{jk}(c, d) \tag{5.2}$$

dove $\{\gamma_{jk}(c, d) \; : \; c = 1, .., C_j \; , \; d = 1, .., C_k \; , \; j \; e \; k = 1, .., M\}$ è un insieme di costanti non negative e dove si pone $\gamma_{jj}(c, d) = 0 \; \forall j, c, d$;

Sia $\underline{x} = (x_1, .., x_j, .., x_M) \in \mathcal{X}$ e $\underline{x}^{'} = (x_1, .., x_j^{'}, .., x_M) \in \mathcal{X}$.

2) se $(x_j, x_j^{'}) \in A_j(c)$, $c = 1, .., C_j$, allora

$$q(\underline{x}, \underline{x}^{'}) = q_j(x_j, x_j^{'}) \nu_j(c) \tag{5.3}$$

dove $\{\nu_j(c) \ : c = 1,..,C_j \ , \ j = 1,..,M\}$ è un insieme di costanti non negative ;

3) se $(x_j^{'}, x_j) \in A_j(c)$ $c = 1,..,C_j$, allora

$$q(\underline{x}, \underline{x}^{'}) = q_j(x_j, x_j^{'})\eta_j(c) \tag{5.4}$$

dove $\{\eta_j(c) \ : c = 1,..,C_j \ , \ j = 1,..,M\}$ è un insieme di costanti non negative ;

4) se $(x_j, x_j^{'}) \not\in U_j$ e $(x_j^{'}, x_j) \not\in U_j$, allora

$$q(\underline{x}, \underline{x}^{'}) = q_j(x_j, x_j^{'}) \tag{5.5}$$

5) infine se $(\underline{x}, \underline{x}^{'})$ non è una transizione possibile

$$q(\underline{x}, \underline{x}^{'}) = 0 \tag{5.6}$$

Diamo ora una condizione sufficiente per la reversibilità della catena di Markov sopra definita .

TEOREMA (5.2)

Se sono verificate le seguenti due ipotesi :

1) il sistema formata dalle (5.7) e dalle (5.8)

$$\alpha_j(c)\ \eta_j(c) = \nu_j(c) \quad ; \quad \forall j = 1,..,M \ , \ \forall c = 1,..,C_j \tag{5.7}$$

$$\alpha_j(c)\ \gamma_{jk}(cd) = \alpha_k(d)\ \gamma_{kj}(dc) \tag{5.8}$$

$$\forall j, k = 1,..,M \ , \ \forall c = 1,..,C_j \ e \ \forall d = 1,..,C_k$$

ha soluzione positiva ;

2) $\forall j$ il processo modificato come segue :

$$\hat{q}(x_j, x_j^{'}) = \alpha_j(c)\ q_j(x_j, x_j^{'}) \ \ ; \quad se (x_j, x_j^{'}) \in A_j(c) \ , \ c = 1,..,C_j$$

$$\hat{q}(x_j, x_j^{'}) = q_j(x_j, x_j^{'}) \ \ ; \quad se (x_j, x_j^{'}) \not\in U_j \tag{5.9}$$

è reversibile ed ha distribuzione all'equilibrio $\hat{\pi}_j(x_j)$;

allora il processo risultante dalla connessione è reversibile ed ha distribuzione all'equilibrio :

$$\pi(\underline{x}) = G^{-1} \prod_{j=1}^{M} \hat{\pi}_j(x_j) \tag{5.10}$$

con G^{-1} costante di normalizzazione.

Dimostrazione

La dimostrazione si trove in [] pag. 881, e non si riporta data la sua lunghezza.

5.4 Una applicazione del teorema di Pollet

Si considera il modello del paragrafo (4.2) e se ne riporta per comodità la struttura. Ogni coda formante la rete è descritta da una catena di Markov $x_i^h(t)$ avente spazio degli stati dato da :

$$\mathcal{X}_i^h = \{0, 1, ..., N_i^h\}\ ; \quad h = 0, ..., M\ ,\ i = 1, ..., F^h$$

con tassi di transizione

$$q(n_i^h, m_i^h) = \lambda_i^h\ 1(n_i^h < N_i^h) \qquad per\ m_i^h = T_{.i}(n_i^h) \quad n_i^h, m_i^h \in \mathcal{X}_i^h$$

$$q(n_i^h, m_i^h) = \mu_i^h(n_i^h) \qquad per\ m_i^h = T_{i.}(n_i^h) \quad n_i^h, m_i^h \in \mathcal{X}_i^h$$

dove si pretende che $\mu_i^h(0) = 0$ e $\mu_i^h(n_i^h) > 0$ per ogni valore assunto da i, h, n_i^h. Per la connessione si fanno le seguenti ipotesi :

1) il routing è Markov dove p_0^h è la probabiltà che un utente, che lascia il trasporto principale, venga indirizzato verso il trasporto locale della h-esima cella $(h = 1, .., M)$,mentre p_i^h rappresenta la probabilità che un utente, lasciante il trasporto della h-esima cella, venga indirizzato verso la coda i della stessa cella $(i = 1, .., F^h)$.

2) il bloccaggio è di tipo 'block and hold'

3) la rete è chiusa ,con N utenti, avente struttura ad albero a collegamenti bidirezionali

Lo spazio di stato della catena di Markov $x(t)$ che descrive l'evoluzione delle rete è

$$\mathcal{X} = \{(n_0^0, .., n_{F^M}^M) \; ; \; n_i^h \in \mathcal{X}_i^h \; , \; h = 0, .., M \; , \; i = 0, .., F^h \; , \; \sum_{h=0}^{M} \sum_{i=0}^{F^h} n_i^h = N\}$$

i tassi di transizione di $x(t)$ sono :

$$q(\underline{n}, \underline{m}) = \mu_0^0(n_0^0) \; p_0^h \; 1(n_0^h < N_0^h) \;\; ; \quad \underline{m} = T^{0h}(\underline{n}) \tag{5.11}$$

$$q(\underline{n}, \underline{m}) = \mu_0^h(n_0^h) \; (1 - \sum_{i=1}^{F^h} p_i^h) \; 1(n_0^0 < N_0^0) \;\; ; \quad \underline{m} = T^{h0}(\underline{n}) \tag{5.12}$$

$$q(\underline{n}, \underline{m}) = \mu_0^h(n_0^h) \; p_c^h \; 1(n_c^h < N_c^h) \;\; ; \quad \underline{m} = T_{0c}^h(\underline{n}) \tag{5.13}$$

$$q(\underline{n}, \underline{m}) = \mu_c^h(n_c^h) \; 1(n_0^h < N_0^h) \;\; ; \quad \underline{m} = T_{c0}^h(\underline{n}) \tag{5.14}$$

dove i simboli sono quelli usati nel paragrafo (4.2) Si vuole verificare la reversibilità della rete e la forma prodotto della sua distribuzione all'equilibrio usando il teorema (5.2). Tali risultati si erano già ottenuti nel paragrafo (4.2) per altra via.

Con riferimento al formalismo di Pollet si definisce :

$$A_i^h = \{(n_i^h, n_i^h + 1) \; : \; n_i^h < N_i^h, n_i^h \geq 0\} \;\; ; \quad h = 0, .., M \; , \; i = 0, .., F^h$$

Per la transizione $T^{0h}(.)$ confrontando la (5.2) con la (5.11) si ottiene :

$$\gamma_{0h}^0 = \frac{p_0^h}{\lambda_0^h} \;\; ; \;\; h = 1, .., M \tag{5.15}$$

Per la transizione $T^{h0}(.)$ confrontando la (5.2) con la (5.12) si ottiene :

$$\gamma_{00}^h = \frac{1 - \sum_{i=1}^{F^h} p_i^h}{\lambda_0^0} \;\; ; \;\; h = 1, .., M \tag{5.16}$$

Per la transizione $T_{0c}^h(.)$ confrontando la (5.2) con la (5.13) si ottiene :

$$\gamma_{0c}^h = \frac{p_c^h}{\lambda_c^h} \;\; ; \;\; h = 1, .., M \;\; c = 1, .., F^h \tag{5.17}$$

Per la transizione $T_{c0}^h(.)$ confrontando la (5.2) con la (5.14) si ottiene :

$$\gamma_{c0}^h = \frac{1}{\lambda_0^h} \;\; ; \;\; h = 1, .., M \;\; c = 1, .., F^h \tag{5.18}$$

Con riferimento al teorema (5.2) tutte le costanti ν e η sono nulle (la rete è chiusa) e quindi la (5.7) è sempre verificata. Riscrivendo la (5.8) per il modello considerato si ha :

$$\alpha_0^h \, \gamma_{00}^h = \alpha_0^0 \, \gamma_{0h}^0 \;\; ; \quad h = i, .., M \tag{5.19}$$

$$\alpha_0^h \, \gamma_{0c}^h = \alpha_c^h \, \gamma_{c0}^h \;\; ; \quad h = i, .., M \;\; c = 1, .., F^h \tag{5.20}$$

Una soluzione non negativa del sistema formato dalle (5.19) e dalle (5.20) è data da : $\alpha_c^h = 1 \; \forall h = 0, .., M \; \forall c = 0, .., F^h$. Con tale soluzione, sostituendo le (5.15) e (5.16) nelle (5.19) si ottiene

$$\lambda_0^0 \, p_0^h = \lambda_0^h \, (1 - \sum_{i=1}^{F^h} p_i^h) \; ; \quad h = 1, .., M \tag{5.21}$$

sostituendo le (5.17) e (5.118) nelle (5.20)

$$\lambda_c^h = \lambda_0^h \, p_c^h \qquad h = 1, .., M \; , \; c = 1, .., F^h \tag{5.22}$$

si sono riottenute cioè le (4.10) e le (4.11).

I processi di Markov modificati, definiti dalla (5.9), sono i processi iniziali ($\alpha_c^h = 1 \; \forall c, h$). Come si è già visto il sistema, formato dalle (5.21) e dalle (5.22) è sempre risolubile, quindi per il teorema (5.2) la catena di Markov $x(t)$ è reversibile e ha distribuzione all'equilibrio (in forma prodotto) data da:

$$\pi(\underline{n}) = G_N \prod_{h=0}^{M} \prod_{i=0}^{F^h} \prod_{r=1}^{n_i^h} \frac{\lambda_i^h}{\mu_i^h(r)} \; ; \;\; \underline{n} \in \mathcal{X}$$

dove gli λ_i^h sono determinati (non univocamente) dalle equazioni (5.21) e dalle equazioni (5.22).

Capitolo 6

CALCOLO DELLA COSTANTE DI NORMALIZZAZIONE

Per reti di code chiuse si pone il problema di calcolare la costante di normalizzazione G_N^{-1}. In questo capitolo viene discussa un'estensione del classico algoritmo di Buzen [1], che permette il calcolo della costante di normalizzazione per reti di code a capacità limitata e particolare attenzione viene data alla valutazione della complessità di calcolo.

6.1 Definizioni e teoremi

Si consideri una rete di code chiusa descritta da una catena di Markov avente spazio di stato :

$$\mathcal{X} = \{(n_1, ..., n_M) \ : \ n_i \in \mathcal{X}_i \ , \ \sum_{i=i}^{M} n_i = N\}$$

$$\mathcal{X}_i = \{0, ..., N_i\}$$

con distribuzione all'equilibrio data da :

$$\pi(\underline{n}) = G_N^{-1} \prod_{i=1}^{M} F_i(n_i) \quad \underline{n} \in \mathcal{X} \tag{6.1}$$

dove

$$F_i(n_i) = \prod_{k=1}^{n_i} f_i(k) \quad i = 1, ..., M$$

$$f_i(k) = \frac{\lambda_i}{\mu_i(k)} \quad i = 1, ..., M$$

e G_N^{-1} è la costante di normalizzazione.

Osservazione : si suppone che la capacità N_i di ogni stazione sia minore o uguale al numero totale di utenti circolanti. Tale ipotesi è , peraltro, non limitativa in quanto in ogni stazione non può contenere più utenti del numero totale N. Si suppone anche ovviamente $\sum_{i=1}^{M} N_i \geq N$

Tale modello può essere, per esempio, quello illustrato nel paragrafo (4.1) o quello illustrato nel paragrafo (4.2) . Si pone ora il problema del calcolo della costante G_N^{-1}.

Osservazione : per reti aperte non si pone il problema del calcolo della costante di normalizzazione, infatti per tali reti la distrubuzione all'equilibrio in forma di prodotto è del tipo :

$$\pi(\underline{n}) = \pi_1(0) * .... * \pi_M(0) \prod_{i=1}^{M} F_i(n_i)$$

dove le $\pi_i(0)$, $i = 1, .., M$, sono date da :

$$\pi_i(0) = \frac{1}{\sum_{j=0}^{N_i} \prod_{k=1}^{j} f_i(k)}$$

e sono facilmente calcolabili data la bassa cardinalità di $\mathcal{X}_i$.

Tornando alle reti chiuse, e sommando la (6.1) su tutti gli stati di $\mathcal{X}$:

$$G_N = \sum_{\underline{n} \in \mathcal{X}} \prod_{i=1}^{M} F_i(n_i) \quad \mathcal{X} \tag{6.2}$$

Per rendersi conto della complessità computazionale del calcolo della costante di normalizzazione mediante la (6.2) è utile il seguente teorema.

TEOREMA (6.1)

L'insieme degli stati $S = \{(n_1, ..., n_M) \; : \; \sum_{i=1}^{M} n_i = N\}$ contiene un numero di stati distinti pari a :

$$\binom{M + N - 1}{N}$$

dimostrazione :

Consideriamo un generico elemento $\underline{n} \in S$. Si tratta di vettore di dimensione M, in relazione biunivoca con la stringa binaria $s(\underline{n})$ data da

$$s(\underline{n}) = append[s(n_1), ..., s(n_M)]$$

dove $s(n_i) = 10......0$, e il numero di zeri presenti è pari a n_i. Ad esempio, per $M = 3$, $N = 5$ si ha che $n = (2, 0, 3)$ è in relazione biunivoca con $s(\underline{n}) = 10011000$, in cui sono presenti M simboli '1' e N simboli zero. Il simbolo '1' in testa a ogni stringa binaria $s(\underline{n})$ non è significativo in quanto sempre presente. Quindi $s(\underline{n})$ può esssere considerata avere $M - 1$ simboli '1' e N simboli zero. Il numero di stringhe binarie distinte è pari alle combinazioni di $N + M - 1$ oggetti (numero totale di simboli) presi ad N ad N. Da cui si ha quanto si voleva dimostrare.

Esempio : Supponendo di avere 10 stazioni e 50 utenti il numero di stati distinti di $S = \{(n_1, ..., n_M) : \sum_{i=1}^{M} n_i = N\}$, per il teorema (6.1), dato da :

$$\binom{M + N - 1}{N} = \frac{59!}{50!\ 9!} \cong 5 \ \cdot \ 10^9$$

Il teorema (6.1) mette quindi in luce che l'equazione (6.2) non è in pratica utilizzabile per il calcolo della costante di normalizzazione, in quanto il numero di stati è estremamente alto anche per modelli di modesta dimensione.

L'algoritmo che viene ora proposto permette il calcolo della costante di normalizzazio per reti di code a capacità limitata.

Si danno alcune definizioni preliminari :

- $S(n, m) = \{(n_1, ..., n_m) \in \mathcal{X}_1 * .... * \mathcal{X}_m : \quad \sum_{i=1}^{m} n_i = n\}$ è lo spazio di stato per la rete ridotta alla connessione delle prime $m \leq M$ stazioni ed in cui circolano $n \leq N$ utenti.

- $g(n, m) = \sum_{\underline{n} \in S(n,m)} \prod_{i=1}^{m} F_i(n_i)$ è la costante di normalizzazione per la rete di code ridotta alla connessione delle prime m stazioni e con n utenti circolanti, e dove si pone, per definizione, $g(0, m) = 1 \ \forall m$.

- $r(n,m) = \frac{g(n-1,m)}{g(n,m)}$ $\forall\ m = 1,..,M\ \forall\ n = 1,..,N$

Osservazione : dalla definizione di $S(n,m)$ e dello spazio di stati $\mathcal{X}$ si ha che :

$$S(N,M) = \mathcal{X}$$

dalla definizione di $g(n,m)$ e dalla (6.2) si ha che :

$$g(N,M) = G_N$$

Definizione : Si definisce inoltre $\pi_m(k,n)$ la probabilità che nella coda m della rete ridotta alle prime m stazioni, con n utenti circolanti, e con sapzio di stato $S(n,m)$, siano presente k utenti .

Di fondamentale importanza per il calcolo del costante di normalizzazione è il seguente teorema.

TEOREMA (6.2)

Valgono le seguenti equazioni

$$\pi_m(0,n) = \pi_m(0,n-1)\frac{r(n,m)}{r(n,m-1)} \tag{6.3}$$

$$\forall \quad n = 1,..,min(N, \sum_{i=1}^{m} N_i) \quad m = 2,..,M$$

$$\pi_m(k,n) = \pi_m(k-1,n-1)\ f_m(k)\ r(n,m) \tag{6.4}$$

$$\forall \quad n = 1,..,min(N, \sum_{i=1}^{m} N_i) \quad m = 1,..,M \quad k = 1,..,min(n, N_m)$$

dove $N_m \leq N\ \forall m = 1,..,M$.

Dimostrazione

Dalla definizione di $g(n,m)$, per $k = 0$, si ha che :

$$\pi_m(0,n) = \sum_{\underline{n}\in S(n,m), n_m=0} \left[\frac{1}{g(n,m)} \prod_{j=1}^{m} F_j(n_j)\right] = \frac{g(n,m-1)}{g(n,m)} =$$

$$= \frac{g(n-1,m-1)}{g(n-1,m)} \cdot \frac{g(n,m-1)}{g(n-1,m-1)} \cdot \frac{g(n-1,m)}{g(n,m)}$$

quindi

$$\pi_m(0,n) = \pi_m(0,n-1)\frac{r(n,m)}{r(n,m-1)}$$

Per $k > 0$ si ha :

$$\pi_m(k,n) = \sum_{\underline{n}\in S(n,m),n_m=k} \left[\frac{1}{g(n,m)} \prod_{j=1}^{m} F_j(n_j)\right] =$$

$$= \frac{1}{g(n,m)} \cdot F_m(k) \cdot \sum_{\underline{n}\in S(n-k,m-1)} \prod_{j=1}^{m-1} F_j(n_j) =$$

$$= \frac{g(n-k,m-1)}{g(n,m)} \cdot F_m(k) =$$

$$= f_m(k) \cdot F_m(k-1) \cdot \frac{g(n-1-(k-1),m-1)}{g(n-1,m)} \cdot \frac{g(n-1,m)}{g(n,m)}$$

quindi

$$\pi_m(k,n) = \pi_m(k-1,n-1)\ f_m(k)\ r(n,m)$$

Osservazione : Nelle formule (6.3) (6.4) compare il fattore comune $r(n,m)$: quindi la conoscenza di $r(n,m)$ non è necessaria al calcolo ricorsivo di $\pi_m(k,n)$, infatti ponendo :

$$\pi^*_m(0,n) = \pi_m(0,n-1)\frac{1}{r(n,m-1)} \tag{6.5}$$

$$n = 1,..,min(N, \sum_{i=1}^{m} N_i) \ \ m = 2,..,M$$

$$\pi^*_m(k,n) = \pi_m(k-1,n-1)f_m(k) \tag{6.6}$$

$$n = 1,..,min(N, \sum_{i=1}^{m} N_i) \ \ m = 1,..,M \ \ k = 1,..,min(n,N_m)$$

sfruttando le (6.3), (6.4), (6.5), (6.6) si ottiene :

$$\pi_m(k,n) = \pi^*_m(k,n)r(n,m) \tag{6.7}$$

$$n = 1,..,min(N, \sum_{i=1}^{m} N_i) \ \ m = 1,..,M \ \ k = 0,..,min(n,N_m)$$

sommando in k la (6.7) si ha che

$$r(n,m) = \frac{1}{\sum_{k=0}^{w} \pi_m^*(k,n)} \tag{6.8}$$

$$n = 1,..,N \quad m = 1,..,M$$

dove $w = min(n, N_m)$.

Osservazione : L'osservazione precedente permette il calcolo della costante $r(n,m)$ (quindi di $\pi_{m+1}(k,n)$) tramite la (6.8). Tale equazione attua una sorta di 'feedback numerico' il quale fa in modo che gli ordini di grandezza delle quantità coinvolte nel calcolo siano tutte uguali. Questo si ripercuote, in modo benefico, sull'errore che si accumula dovuto al troncamento delle quantità reali (rappresentate in virgola mobile)

TEOREMA (6.3)
Si verifica che :

$$G_N = \prod_{n=1}^{N} \frac{1}{r(n,M)} \tag{6.9}$$

Dimostrazione :
Dalla definizione di $r(n,M)$ si ha :

$$\prod_{n=1}^{N} \frac{1}{r(n,M)} = \frac{g(1,M)}{g(0,M)} \cdot \frac{g(2,M)}{g(1,M)} \cdot \cdot \cdot \cdot \frac{g(N,M)}{g(N-1,M)}$$

poichè $g(0,M) = 1$ per definizione, si ottiene direttamente la (6.9).

6.2 Algoritmo per il calcolo di G_N

In base alle definizioni e ai teoremi dati nel paragrafo precedente si può scrivere in forma astratta l'algoritmo che calcola la costante di normalizzazione :

- inizializza m=1
- calcola $r(n,1) = \frac{g(n-1)}{g(n,1)} = \frac{1}{f_1(n)}$
- ripeti finchè m=M
-inizio

-m=m+1

-calcola $\pi^*_m(K, M)$ con la (6.6)

-calcola $r(n, m)$ con la (6.8)

-calcola $\pi_m(K, M)$ con la (6.7)

-fine

- calcola G_N con la (6.9)

Osservazione : L'algoritmo proposto non calcola solo la costante di normalizzazione ma anche le probabilità $\pi_M(k, N)$ di avere k, $k = 1, .., n_M$, utenti nella coda M-esima della rete originale (non ridotta). Scambiando M-1 volte fra loro le code, e ripetendo l'algoritmo, si possono ottenere le probabilità $\pi(k, N)$ per tutte le code.

Entrando più nel dettaglio si fornisce l'algoritmo in pseudo codice. Si indicherà $f_i(k)$ con f_i(k) e N_i con N_i.

```
m:1..M                          ( stazione m )
n:1..N                          ( utente   n )
k:1..N                          ( variabile ausiliaria )
Max_m                           ( massimo numero di utenti
                                  ospitabili nella rete al
                                  passo m )
Prec_max                        ( massimo numero di utenti
                                  ospitabili nella rete al
                                  passo precedente m-1 )
p array [0,..,N]                ( vettore delle probabilita')
r array [0,..,N]                ( vettore delle costanti r(n,m) )
N_i                             ( capacita' della coda i )

begin
 m=1
 prec_max =N_1
 for n=1 to N_1 do
```

```
  r(n)=1/f_1(n)                        ( per definizione di r(n,m) )
 repeat
  begin
   m=m+1
   Max_m=min(N,prec_max + N_m)
   p[0]=1
   for n=1 to Max_m do
    begin
     sum=0
     kk=min(N_m,n)                     ( rappresenta il massimo
                                         numero di utenti ospitabili
                                         nella stazione m )
     for k=kk downto 1 do
      begin
       p[k]=p[k-1] * f_m(k)            ( implementa la (6.6) )
       sum = sum + p[k]
      end
     if n > prec_max                   ( la stazione m deve allora
                                         ospitare almeno un utente )
      then
       p[0]=0
      else
       pi(0)=p[0] / r(n)               ( implementa la (6.5) )
     r(n)=1 / [ sum + p[0] ]           ( implementa la (6.8) )
     for k=0 to kk do
      p[k]=p[k] * r(n)$                ( implementa la (6.7) )
    end
   prec_max=Max_m
  end
 until m=M
end
```

L'algoritmo sopra esposto richiede il seguente numero di somme :

$$\sum_{m=2}^{M}\left[\sum_{n=1}^{Max\_m} min(N_m, n) + (Max\_m + 1)\right] \tag{6.10}$$

e il seguente numero di moltiplicazioni-divisioni

$$\sum_{m=2}^{M}\left[\sum_{n=1}^{Max\_m} min(N_m, n) + \frac{Max\_m(Max\_m + 5)}{2} + prec\_max\right] + N_1 \tag{6.11}$$

dove

$$Max\_m = min(N, \sum_{i=1}^{m} N_i)$$

La (6.10) e la (6.11) sono massime per reti di code a capacità non limitata ($N_i = N \ \forall i = 1, .., M$), in questo caso la (6.10) diventa :

$$\sum_{m=2}^{M}\left[\sum_{n=1}^{N} n + (N + 1)\right] = \frac{(M-1)(N+1)(N+2)}{2} \tag{6.12}$$

mentre la (6.11) diventa :

$$\sum_{m=2}^{M}\left[\sum_{n=1}^{N} n + \frac{N(N+5)}{2} + N\right] + N = (M-1)[(N+1)^2 + 1] + N \tag{6.13}$$

Quindi per reti di code a capacità non limitata, M ed N grandi l'algoritmo richiede un numero di somme pari a circa $(M \ N^2)/2$ ed un numero di moltiplicazioni divisioni pari a circa $M \ N^2$

Riprendendo l'esempio del paragrafo precedente, M=10 e N=50 la (6.12) fornisce :

$$\frac{(M-1)(N+1)(N+2)}{2} = 9 \cdot 51 \cdot 26 \cong 12 \cdot 10^3$$

In realtà le (6.12) e le (6.13) sono approssimazioni molto larghe per reti a capacità limitata, per tali reti il numero di somme e prodotti-divisioni sono notevolmente inferiori.

6.3 Il codice in appendice

In appendice è stato riportato un programma che implementa il modello del paragrafo (4.2). Il programma richiede in input :

- il numero di utenti N
- il numero di celle h
- il numero di stazioni serventi, F^h, per ogni cella
- le capacità N_i^h
- le probabilità di transizione p_i^h
- i tassi di servizio $\mu_i^h(r)$

dove tutti i simboli utilizzati sono definiti nel paragrafo (4.2).

Osservzione : Si noti che

$$\sum_{h=1}^{M} p_0^h = 1$$

poichè un utente, che lascia il trasporto principale, non può che andare in uno degli M trasporti locali, inoltre :

$$\sum_{i=1}^{F^h} p_i^h \leq 1$$

poichè non può essere nulla la probabilità che un utente passi dal trasporto locale h al trasporto principale.

L'algoritmo usato è quello illustrato nel paragrafo precedente, e il programma fornisce in uscita per ogni coda:

- le probabilità $\pi(k, N)$
- il throughput
- l'utilizzazione
- la lunghezza media
- tempo medio di residenza
- la costante di normalizzazione

dove

- il throughput è il numero medio di utenti che transitano nell'unità di tempo nella coda considerata
- l'utilizzazione è il rapporto tra throughput e il massimo rateo di servizio.

Si riporta un esempio su cui è stato testato il programma.

Test 1

Si considera una struttura del tipo del paragrafo (4.2), con un trasproto principale, un trasporto locale e due stazioni serventi. Tutte le stazioni hanno capactà unitaria e nella rete sono presenti due utenti. Con p_1, p_2 si indicano le probabiltà di passare dal trasporto locale ad una delle due stazioni serventi. Si risolve analiticamente il modello e si confrontano i risultati con i risultati del programma.

Sono presenti $\binom{4}{2} = 6$ stati differenti :

(1,2) stato 1 (un utente nella stazione 1 e un utente nella stazione 2)

(1,3) stato 2 (un utente nella stazione 1 e un utente nella stazione 3)

(1,4) stato 3 (un utente nella stazione 1 e un utente nella stazione 4)

(2,3) stato 4 (un utente nella stazione 2 e un utente nella stazione 3)

(2,4) stato 5 (un utente nella stazione 2 e un utente nella stazione 4)

(3,4) stato 6 (un utente nella stazione 3 e un utente nella stazione 4)

Le possibili transizioni sono : $1 \leftrightarrow 2$, $1 \leftrightarrow 3$, $2 \leftrightarrow 4$, $3 \leftrightarrow 5$, $4 \leftrightarrow 6$.

Utilizzando l'equazione (2.1) per le possibili transizioni e imponendo che la somma delle probabilità valga uno si ottiene il sistema di sei equazioni in sei incognite :

$$\pi_1 \mu_2 p_1 = \pi_2 \mu_3$$

$$\pi_1 \mu_2 p_2 = \pi_3 \mu_4$$

$$\pi_2 \mu_1 = \pi_4 \mu_2 (1 - p_1 - p_2)$$

$$\pi_3 \mu_1 = \pi_5 \mu_2 (1 - p_1 - p_2)$$

$$\pi_4 \mu_2 p_2 = \pi_6 \mu_4$$

$$\sum_{i=1}^{6} \pi_i = 1$$

esplicitando rispetto a π_1 si ha che :

$\pi_1 = \left[1 + \frac{\mu_2 p_1}{\mu_3} + \frac{\mu_2 p_2}{\mu_4} + (1 + \frac{\mu_2 p_2}{\mu_4}) \frac{\mu_2 p_1}{\mu_3} \frac{\mu_1}{\mu_2 (1 - p_1 - p_2)} + \frac{\mu_2 p_2}{\mu_4} \frac{\mu_1}{\mu_2 (1 - p_1 - p_2)} \right]$ Una volta calcolato π_1 è immediato calcolare le altre incognite. Indicando con $\pi_i(1,2)$ la probabilità di avere un utente nella coda i-esima essendo presenti due utenti nella rete si ha che

$$\pi_1(1,2) = \pi_1 + \pi_2 + \pi_3$$

$$\pi_2(1,2) = \pi_1 + \pi_4 + \pi_5$$

$$\pi_3(1,2) = \pi_2 + \pi_4 + \pi_6$$

$$\pi_4(1,2) = \pi_3 + \pi_5 + \pi_6$$

Con $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 1$ e $p_1 = \frac{1}{4}, p_2 = \frac{1}{2}$ si ottiene :

$$\pi_1(1,2) = \frac{7}{21} \qquad \pi_2(1,2) = \frac{16}{21}$$

$$\pi_3(1,2) = \frac{7}{21} \qquad \pi_4(1,2) = \frac{12}{21}$$

Il programma fornisce in output :

```
RISULTATI DELLA STAZIONE '1' (buffer=1) :


prob(0)=0.666667
prob(1)=0.333333


SOMMA DELLE PROBABILITA'=1 , COSTANTE DI NORMALIZZAZIONE=0.047619


THROUGHPUT   =0.333333    LUNGHEZZA MEDIA=0.333333
UTILIZZAZIONE=0.333333    TEMPO MEDIO    =1


RISULTATI DELLA STAZIONE '2' (buffer=1) :


prob(0)=0.238095
prob(1)=0.761905


SOMMA DELLE PROBABILITA'=1 , COSTANTE DI NORMALIZZAZIONE=0.047619


THROUGHPUT   =0.761905    LUNGHEZZA MEDIA=0.761905
UTILIZZAZIONE=0.761905    TEMPO MEDIO    =1


RISULTATI DELLA STAZIONE '3' (buffer=1) :
```

```
prob(0)=0.666667
prob(1)=0.333333

SOMMA DELLE PROBABILITA'=1 , COSTANTE DI NORMALIZZAZIONE=0.047619

THROUGHPUT   =0.333333    LUNGHEZZA MEDIA=0.333333
UTILIZZAZIONE=0.333333    TEMPO MEDIO    =1

RISULTATI DELLA STAZIONE '4' (buffer=1) :

prob(0)=0.428571
prob(1)=0.571429

SOMMA DELLE PROBABILITA'=1 , COSTANTE DI NORMALIZZAZIONE=0.047619

THROUGHPUT   =0.571429    LUNGHEZZA MEDIA=0.571429
UTILIZZAZIONE=0.571429    TEMPO MEDIO    =1
```

Il programma è stato testato per numerosi altri valori dei parametri $p_1, p_2, \mu_1, \mu_2, \mu_3, \mu_4$ fornendo sempre valori corretti.

Test 2

Si è considerato un modello con un trasporto principale, due trasporti locali e due stazioni serventi per ogni trasporto locale, con un solo utente circolante (sette equazioni in sette incognite). Il programma ha fornito valori corretti per vari valori dei parametri.

Le varie verifiche sono state fatte, come si è visto, per modelli molto semplici. Questo perché complicando anche di poco i modelli si va incontro a conti di notevole dimensione. Per esempio se nel test 1 si considera il modesto numero di cinque utenti, con capacità di tutte le stazioni uguale a cinque, si ottengono 56 equazioni in 56 inconite.

Per modelli di grosse dimensioni sono stati fatti i seguenti riscontri :

- le probabilità si sommano sempre a uno
- per reti di code aventi capacità massima uguale a N + k e con N utenti circolanti, per ogni coda solo le probabilità comprese tra la probabilità di saturazione e la probabilità di saturazione meno k sono diverse da zero.
- per strutture simmetriche si hanno probabilità simmetrice.

Il programma fatto girare su un Pentium a 133 MHz a 16 Mbyte con 100 utenti e 25 stazioni fornisce i risultati di una stazione in un tempo inferiore ad un secondo.

Conclusioni

In questa tesi si sono studiate le reti di code a capacità limitata con struttura ad albero. Se ne dimostra la reversibilità e la forma prodotto della distribuzione all'equilibrio, sotto le ipotesi che le singole code siano modellabili mediante catene di Markov, il routing sia di Markov, i collegamenti fra code adiacenti siano bidirezionali, inoltre per reti aperte si suppone che gli utenti possano entrare (dall'esterno) o uscire (verso l'esterno) attraverso una sola coda. Si sono ottenuti questi risultati utilizzando la teoria della reversibilità di Kelly [7]. Per reti a struttura stellare composta (caso particolare del precedente) si sono ottenuti gli stessi risultati anche per altra via utilizzando cioè il teorema di Pollet [13]. Per queste reti è stato implementato un codice che calcola le distribuzioni di probabilità marginali e la costante di normalizzazione della rete, utilizzando l'algoritmo di Buzen [1] per reti a capacità limitata con distribuzione all'equilibrio in forma di prodotto. La validazione è stata effettuata, per modelli di piccole dimensioni, tramite calcolo manuale. Per modelli di grande dimensione tramite 'controlli di necessità'.

Appendice

```
class stazioni_serventi
{
 public:
  char y;                       // Ratei dipendenti dallo stato ?.
  int Cap;                      // Capacita' della stazione.
  long double tasso_arrivo;     // Tasso di arrivo.
  long double * mu;             // Vettore ratei di servizio.
  stazioni_serventi(int,char); // Costruttore.
  ~stazioni_serventi();         // Distruttore.
};

class trasporto_locale
{
 public:
  char y;
  int  Cap;
  int  num_staz_serv;          // Numero di stazioni serventi collegate.
  long double tasso_arrivo;
  long double * mu;

  long double * p;             /* Vettore di probabilita' di transizione
                                  di un utente dal trasporto locale a
                                  una stazione servente.
                                  La somma di queste probabilita' deve
```

```
                                  essere minore di uno.                  */

  stazioni_serventi * * t;    /* Vettore di puntatori alle stazioni
                                 serventi collegate  al  trasporto
                                 locale in questione.                  */



  trasporto_locale(int ,int ,char );
  ~trasporto_locale();
};

class trasporto_principale
{
 public:
  char y;
  int Cap;
  int num_tras_loc;           // Numero trasporti locali collegati.
  long double tasso_arrivo;
  long double * mu;

  long double * p;            /* Vettore di probabilita' di transizione
                                 di un utente dal trasporto principale
                                 ad un suo trasporto locale.
                                 La somma di queste probabilita deve
                                 essere unitaria .    */

  trasporto_locale * * t;     // Vettore di puntatori ai trasporti locali.

  trasporto_principale(int ,int ,char );
  ~trasporto_principale();
};
```

```
  class FMS
  {
   public:
    trasporto_principale * t; // Puntatore al trasporto principale.
    int num_utenti;           // Numero di utenti nella rete.
    int num_tot_code;         // Numero totale code della rete.
    FMS() { t=NULL;};
    ~FMS() { delete t; };
  };

// Prototipi delle funzioni.

int  min(int,int);
int  posti_ok(int,int,int **);
int  buffer_ok(int**,int,int);
int  c_break(void);
long double throughput(long double*,int,char,long double * );
long double max(long double *,int,char);
long double lunghezza_med(long *,int);
void Leggi_FMS(FMS &);
void Tassi_arrivi(FMS &);
void iniz_Buzacott(FMS &,long double **,long double **,int **,char **);
void Buzacott(FMS &,long double **,long double **,int **,char **,int);
void scambia_long(int,int,long double * *);
void scambia_int(int,int,int * *);
void scambia_char(int,int,char * *);
void stampa(long double *,long double &,long double &,long double &,
                            long double &, long double &,int &,int &);
#include <iostream.h>
#include "fms.h"
#include <conio.h>
#define ABORT 0
```

```
//  IMPLEMENTAZIONE FUNZIONI.



//  COSTRUTTORE DELLA CLASSE STAZIONI_SERVENTI.

stazioni_serventi::stazioni_serventi(int capacit,char x)
{
  y=x;
  Cap=capacit;
  if(y=='s')
   {
    mu=new long double[capacit + 1];
    for(int i=0;i<=capacit;++i)
     mu[i]=0;
   }
  else
   {
    mu=new long double[2];
    mu[1]=0;
   }
};



//  DISTRUTTORE DELLA CLASSE STAZIONI_SERVENTI.

stazioni_serventi::~stazioni_serventi() { delete mu;  };
```

```
// COSTRUTTORE DELLA CLASSE TRASPORTO_LOCALE.

trasporto_locale::trasporto_locale(int capacit,int num_staz,char x)
 {
   y=x;
   Cap=capacit;
   num_staz_serv=num_staz;
   p=new long double [num_staz + 1];
   t=new stazioni_serventi*[num_staz + 1];
   for(int i=0;i<=num_staz;++i)
  {
    t[i]=NULL;
    p[i]=0;
  }
 if (y=='s')
  {
    mu=new long double[capacit + 1];
    for(i=0;i<=capacit;++i)
     mu[i]=0;
  }
 else
  {
    mu=new long double[2];
    mu[1]=0;
  }
};


// DISTRUTTORE DELLA CLASSE TRASPORTO_LOCALE.

trasporto_locale::~trasporto_locale()
```

```
 {
  delete p ;
  delete mu;
  for(int i=1;i<=num_staz_serv;++i)
   delete t[i] ;
 };


// COSTRUTTORE DELLA CLASSE TRASPORTO_PRINCIPALE.

trasporto_principale::trasporto_principale(int capacit,int num_tras,char x)
{
 y=x;
 Cap=capacit;
 num_tras_loc=num_tras;
 p=new long double[num_tras + 1];
 t=new trasporto_locale*[num_tras + 1];
 for(int i=0;i<=num_tras;++i)
  {
    t[i]=NULL;
    p[i]=0;
  }
 if (y=='s')
  {
    mu=new long double[capacit + 1];
    for(i=0;i<=capacit;++i)
     mu[i]=0;
  }
 else
  {
    mu=new long double[2];
    mu[1]=0;
```

```
  }
};

// DISTUTTORE DELLA CLASSE TRASPORTO_PRINCIPALE.

trasporto_principale::~trasporto_principale()
 {
  delete p ;
  delete mu;
  for(int i=1;i<=num_tras_loc;++i)
   delete t[i] ;
 };



/* FUNZIONE DI LETTURA DEI RATEI DI SERVIZIO DI UNA CODA ,
    h : CAPACITA' DELLA CODA
    s : RATEI DIPENDENTI DALLO STATO ? (s,n)
    c : VETTORE RATEI                                       */

void Leggi_mu(char s,int h,long double * c)
{
 if (s=='n')
  {
   cout<<"IMMETTI IL RATEO mu :";
   cin>>c[1];
  }
 else
  {
   cout<<"IMMETTI I RATEI mu(k) k=1,..,"<< h <<":"<< endl <<endl;
   for(int k=1;k<=h;++k)
    cin>>c[k];
  }
```

```
 cout<<endl;
};



/* FUNZIONE DI LETTURA DELLE PROBABILITA' DI TRASNIZIONE DI UN  UTENTE
   DAL TRASPORTO PRINCIPALE AD UN TRASPORTO LOCALE O DA UN TRASPORTO
   LOCALE AD UN SUO TRASPORTO LOCALE.NEL PRIMO CASO c(0)=0,NEL SECONDO
   c(0) E' UGUALE ALLA PROBABILITA'CHE UN UTENTE , LASCIANTE UN TRASPORTO
   LOCALE , VADA NEL TRASPORTO PRINCIPALE.

    b : NUMERO DI TRASPORTI LOCALI O DI STAZIONI SERVENTI
COLLEGATE ALLA CODA CHIAMANTE.
    c : VETTORE PROBABILITA'.                                        */

void Leggi_p(int b,long double * c)
{
 cout<<"PROBABILIA' DI TRANSIZIONE p(i) i=1,..,"<<b<<":"<<endl<<endl;
 c[0]=0;
 for(int i=1;i<=b;++i)
  {
   cin>>c[i];
   c[0]+=c[i];
  };
 int ok=1;
 for(i=1;i<=b;++i)
   if( !( c[i]>0 & c[i]<=1 ) )
    { ok=0 ; i=b; }
 if( !ok )
  {
   cout<<endl;
   cout<<"DEVONO ESSERE IN (0,1] REIMMETTI LE ";
   Leggi_p(b,c);
```

```
  }
 else{c[0]=1-c[0];}
};


// FUNZIONE DI LETTURA DI TUTTI I DATI DELLA RETE .

void Leggi_FMS(FMS & rete)
{
 int temp_int1,temp_int2,temp_int3;
 char temp_char;

 clrscr();
 cout<<endl;
 cout<<"IMMETTI IL NUMERO DI UTENTI DELLA RETE (chiusa) :";
 cin>>temp_int1;
 cout<<endl;
 rete.num_utenti=temp_int1;

 cout<<"IMMETTI LA CAPACITA’ DEL TRASPORTO PRINCIPALE :";
 cin>>temp_int1;
 cout<<endl;

 cout<<"I RATEI DEL TRASORTO PRINCIPALE SONO FUNZIONE DELLO STATO ? (s/n) :";
 rimmetti_3:
 cin>>temp_char;
 if( temp_char!=’n’ && temp_char!=’s’ )
  {
   cout<<endl;
   cout<<"(s,n) :";
   goto rimmetti_3;
  }
```

```
cout<<endl;

cout<<"IMMETTI IL NUMERO DI TRASPORTI LOCALI : ";
cin>>temp_int2;
cout<<endl;

rete.num_tot_code=temp_int2 + 1;

rete.t=new trasporto_principale(temp_int1,temp_int2,temp_char);

cout<<"PER IL TRASPORTO PRINCIPALE , ";
Leggi_mu(temp_char,temp_int1,(*(rete.t)).mu);

cout<<"PER IL TRASPORTO PRINCIPALE , ";
rimmetti_1:
Leggi_p(temp_int2,(*(rete.t)).p);
cout<<endl;
if( (((*(rete.t)).p)[0]) )
 {
  cout<<"                    LA SOMMA DEVE ESSERE UNITARIA "<<endl;
  cout<<"           RIMMETTI LE ";
  goto rimmetti_1;
 }

temp_int3=temp_int2;
for(int i=1;i<=temp_int3;++i)
 {
  cout<<"IMMETTI LA CAPACITA' DEL TRASPORTO LOCALE '"<< i <<"' :";
  cin>>temp_int1;
  cout<<endl;

  cout<<"I RATEI DEL TRASPORTO LOCALE '"<<i<<"' SONO FUNZIONE DELLO STATO ? (s/n) :";
```

```
rimmetti_4:
cin>>temp_char;
if( temp_char!='n' && temp_char!='s' )
 {
  cout<<endl;
  cout<<"(s,n) :";
  goto rimmetti_4;
 }
cout<<endl;

cout<<"IMMETTI IL NUMERO DI STAZIONI SERVENTI DEL TRASPORTO LOCALE '";
cout<<i<<"' :";
cin>>temp_int2;
cout<<endl;

rete.num_tot_code+=temp_int2;

((*(rete.t)).t)[i]=new trasporto_locale(temp_int1,temp_int2,temp_char);

cout<<"PER IL TRASPORTO LOCALE '"<<i<<"' ";
Leggi_mu(temp_char,temp_int1,(*(((*(rete.t)).t)[i])).mu);

cout<<"PER IL TRASPORTO LOCALE '"<<i<<"' ";
rimmetti_2:
Leggi_p(temp_int2,(*(((*(rete.t)).t)[i])).p);
cout<<endl;
if( ((*(((*(rete.t)).t)[i])).p)[0] <= 0  )
 {
  cout<<"                     LA SOMMA DEVE ESSERE MINORE DI UNO "<<endl;
  cout<<"          RIMMETTI LE ";
  goto rimmetti_2;
 }
```

```
   for(int k=1;k<=temp_int2;++k)
    {
     cout<<"IMMETTI LA CAPACITA' DELLA STAZIONE SERVENTE '"<<k<<"'";
     cout<<" DEL TRASPORTO LOCALE '"<<i<<"' :";
     cin>>temp_int1;
     cout<<endl;

     cout<<"  I RATI DELLA STAZIONE SERVENTE '"<<k<<"' APPARTENENTE AL ";
     cout<<"TRASPORTO LOCALE '"<<i<<"'"<<endl ;
     cout<<"                SONO DIPENDENTI DALL STATO ? (s/n):";
     rimmetti_5:
     cin>>temp_char;
     if( temp_char!='n' && temp_char!='s' )
      {
       cout<<endl;
       cout<<"(s,n) :";
       goto rimmetti_5;
      }
     cout<<endl;

     ((*(((*(rete.t)).t)[i])).t)[k]=new stazioni_serventi(temp_int1,temp_char);

     cout<<"   PER LA STAZIONE SERVENTE '"<<k<<"' APPARTENENTE AL ";
     cout<<"TRASPORTO LOCALE '"<<i<<"' "<<endl;
     cout<<"                     ";
     Leggi_mu(temp_char,temp_int1,(*(((*(((*(rete.t)).t)[i])).t)[k])).mu);
    };
  };
};
```

```
// FUNZIONE CHE CALCOLA I TASSI DI ARRIVO DI TUTTE LE CODE.

void Tassi_arrivi( FMS & rete)
 {
  const long double tass_arriv_princ = 1;

  trasporto_locale * loc_k;       // Puntatore al trasporto locale k-esimo.
  trasporto_principale * princ;   // Puntatore al trasporto principale.
  stazioni_serventi * staz_k_i;   // Puntatore alla stazione servente
  // 'i' del trasport locale 'k'.
  princ=rete.t;
  princ->tasso_arrivo=tass_arriv_princ;
  int tras_loc=princ->num_tras_loc;
  for(int k=1; k<=tras_loc ; ++k )
   {
    loc_k=((*(rete.t)).t)[k];
    loc_k->tasso_arrivo=tass_arriv_princ * ((princ->p)[k]) / (loc_k->p)[0];
    for(int i=1 ; i<=loc_k->num_staz_serv; ++i )
     {
      staz_k_i=(loc_k->t)[i];
      staz_k_i->tasso_arrivo= (loc_k->tasso_arrivo)*(loc_k->p)[i];
     };
   };
 };


/* FUNZIONE DI INIZIALIZZAZIONE  DELL'ALGORITMO  CHE CALCOLA LE PROBABILITA'
   DI AVERE k UTENTI IN UNA CODA DELLA RETE.

   TUTTI I VETTORI DI PUNTATORI HANNO DIMENSIONE UGUALE AL NUMERO DI CODE ,
   LA PRIMA COMPONENTE DI OGNI VETTORE SI RIFERISCE AL TRASPORTO PRINCIPALE
   LA  SECONDA AL PRIMO  TRASPORTO LOCALE , LA TERZA  ALLA  PRIMA  STAZIONE
```

```
   SERVENTE DEL PRIMO TRASPORTO LOCALE, ..... ,L'ULTIMA ALL'ULTIMA STAZIONE
   SERVENTE DELL'ULTIMO TRASPORTO LOCALE.  */

void iniz_Buzacott(FMS & rete,long double * * tass_arrivi,long double
    ** tass_servizi,int ** cap,char ** z)
{
 trasporto_principale * princ=rete.t;
 trasporto_locale * loc_k;
 stazioni_serventi * staz_k_i;
 int tras_loc=princ->num_tras_loc;
 z[1]=&(princ->y);
 cap[1]=&(princ->Cap);
 tass_arrivi[1]=&(princ->tasso_arrivo);
 tass_servizi[1]=princ->mu;

 int l=0;
 int i;
 for(int k=1;k<=tras_loc;++k)
  {
   loc_k=princ->t[k];
   z[k+l+1]=&(loc_k->y);
   cap[k+l+1]=&(loc_k->Cap);
   tass_arrivi[k+l+1]=&(loc_k->tasso_arrivo);
   tass_servizi[k+l+1]=loc_k->mu;
   i=1;
   while(i<=loc_k->num_staz_serv)
    {
     staz_k_i=(loc_k->t)[i];
     z[k+l+i+1]=&(staz_k_i->y);
     cap[k+l+i+1]=&(staz_k_i->Cap);
     tass_arrivi[k+l+i+1]=&(staz_k_i->tasso_arrivo);
     tass_servizi[k+l+i+1]=staz_k_i->mu;
```

```
     i=i+1;
    };
   l=l+i-1;
  };
};



// CALCOLA IL MINIMO FRA DUE INTERI.

int min(int a , int b)
 {
  if(a<b){ return a; }
  else   { return b; }
 };



// CALCOLA  IL THROUGHPUT DI DI UNA CODA NEI CASI FCFS , LCFS , PS.

long double throughput(long double * prob_1,int cap,char s,long double * mu)
 {
  int i;
  long double somma=0;
  long double throu=0;
  if(s=='n')
   {
    for(i=1;i<=cap;++i)
     somma +=prob_1[i];
    throu=somma*mu[1];
    return throu;
   }
  for(i=1;i<=cap;++i)
   throu += prob_1[i]*mu[i];
```

```
  return throu;
 };


// CALCOLA IL MASSIMO DEGLI ELEMENTI DI UN VETTORE.

long double max(long double * mu,int cap,char s)
 {
  long double max =0;
  if(s=='n') return mu[1];
  for(int i=1;i<=cap;++i)
   if(mu[i]>max)
    max=mu[i];
  return max ;
 };


// CALCOLA  LA LUNGHEZZA MEDIA DI UNA CODA.

long double lunghezza_med(long double * prob_1,int cap)
 {
  long double lungh=0;
  for(int i=1;i<=cap;++i)
   lungh += i* prob_1[i];
  return lungh;
 };


/* FUNZIONE CHE CACOLA,CON L'ALGORITMO DI BUZZACOTT , LA PROBABILITA' DI
   AVERE K UTENTI IN UNA CODA DELLA RETE.

    tassi_arrivi  : VETTORE DI PUNTARI (dimensione=numero code) AI TASSI
```

```
    DI ARRIVO.
    tassi_servizi : VETTORE DI PUNTARI (dimensione=numero code) AI RATEI
    DI SERVIZIO.
    buffer_code   : VETTORE DI PUNTARI ( dimensione = numero code ) ALLE
    CAPACITA' DELLE SINGOLE CODE.
    carattere     : FORNISCE INDICAZIONE SULLA DIPENDENZA , O MENO , DEI
    TASSI DI SERVIZIO DAL NUMERO DI UTENTI IN CODA.
*/

void Buzacott(FMS & rete,long double * * tassi_arrivi,long double
       ** tassi_servizi,int ** buffer_code,char ** carattere,int ind)
{
 long double * prob=new long double[rete.num_utenti + 1];
 long double * r=new long double[rete.num_utenti + 1];
 int m,n,k,Nm,prec_Nm,kk;
 long double sum;

 m=1;
 prec_Nm=*(buffer_code[1]);
 if(*(carattere[1])=='n')
  {r[0]=( (tassi_servizi[1])[1] ) / ( *(tassi_arrivi[1]) ) ;
   for(n=1;n<=prec_Nm;++n)
    r[n]=r[0];
  }
 else
  {
   for(n=1;n<=prec_Nm;++n)
    r[n]=( (tassi_servizi[1])[n] ) / ( *(tassi_arrivi[1]) );
  }
 do
  {
   m=m+1;
```

```
   Nm=min(rete.num_utenti,prec_Nm + (*(buffer_code[m])) );
   prob[0]=1;
   for(n=1;n<=Nm; ++ n )
    {
     sum=0;
     kk=min(n,*(buffer_code[m]));
     if( *(carattere[m])=='n')
      {
       for( k=kk; k != 0; --k)
{
 prob[k]=prob[k-1] * (*(tassi_arrivi[m])) / (tassi_servizi[m])[1];
 sum=sum + prob[k];
};
      }
     else
      {
       for( k=kk; k != 0; --k)
{
 prob[k]=prob[k-1] * ( (*(tassi_arrivi[m])) / (tassi_servizi[m])[k] );
 sum=sum + prob[k];
};
      }
     if(n>prec_Nm){prob[0]=0;}
     else{prob[0]=prob[0]/r[n];}
     r[n]=1/(sum + prob[0]);
     for(k=0;k<=kk;++k)
      prob[k]=(prob[k]) * (r[n]);
    };
   prec_Nm=Nm;
  } while(m<rete.num_tot_code);

 long double G;
```

```
 G=1;
 for(n=1; n<=rete.num_utenti ; ++n)
   G=G*r[n];

 long double thr,utilizzazione,lungh_med,tempo_med;
 thr=throughput(prob,*(buffer_code[m]),*(carattere[m]),tassi_servizi[m]);
 utilizzazione=thr/max(tassi_servizi[m],*(buffer_code[m]),*(carattere[m]));
 lungh_med=lunghezza_med(prob,*(buffer_code[m]));
 tempo_med=lungh_med/thr;

 stampa(prob,thr,utilizzazione,lungh_med,tempo_med,G,*(buffer_code[m]),ind );

 delete r;
 delete prob;
};


// SCAMBI I PUNTATORI 'i' E 'k' DI UN VETTORE DI PUNTATORI A LONG DOUBLE.

void scambia_long(int i,int k,long double ** s)
 {
  long double * temp;
  temp=s[i];
  s[i]=s[k];
  s[k]=temp;
 };


// SCAMBI I PUNTATORI 'i' E 'k' DI UN VETTORE DI PUNTATORI A INT.

void scambia_int(int i,int k,int ** s)
 {
```

```
  int * temp;
  temp=s[i];
  s[i]=s[k];
  s[k]=temp;
 };


// SCAMBI I PUNTATORI 'i' E 'k' DI UN VETTORE DI PUNTATORI A CHAR.

void scambia_char(int i,int k,char ** s)
 {
  char * temp;
  temp=s[i];
  s[i]=s[k];
  s[k]=temp;
 };

// CONTROLLA SE LA RETE PUO' CONTENERE IL NUMERO DI UTENTI.

int posti_ok(int numero_code,int numero_utenti,int ** capacit)
 {
  int somma=0;
  for(int i=1;i<=numero_code;++i)
   somma += (*(capacit[i])) ;
  if(somma<numero_utenti)
   return 0;
  return 1;
 };

/* VERIFICA CHE LE CAPACITA' SIANO MINORI O UGUALI DEL NUMERO
   DI UTENTI PRESENTI NELLA RETE.
   TALE CONDIZIONE NON IMPONE RESTIZIONI ALL'INSIEME DI RETI
```

```
   PROCESSABILI */

int buffer_ok(int ** cap,int numero_code,int numero_utenti)
 {
  for(int i=1;i<=numero_code;++i)
   if( *(cap[i])>numero_utenti )
    return 0;
  return 1 ;
 };

// STAMPA I RISULTATI A VIDEO. E'INVOCATA NELLA FUNZIONE BUZACOTT.

void stampa(long double * prob_1,long double & throu,
    long double & utiliz,long double & lungh,
    long double & tempo,long double & const_norm,
    int & cap,int & indice)
{
 clrscr();
 cout<<"RISULTATI DELLA STAZIONE '"<<indice<<"' (buffer="<<cap<<") :"<<endl;
 cout<<endl;
 for(int i=0 ;i<=cap;++i)
  cout<<"("<<i<<")="<<prob_1[i]<<endl;
 cout<<endl;
 long double sum=0;
 for(i =0;i<=cap;++i)
  sum +=prob_1[i];
 cout<<"SOMMA DELLE PROBABILITA'="<<sum;
 cout<<" , COSTANTE DI NORMALIZZAZIONE="<<const_norm <<endl<<endl;
 cout<<"THROUGHPUT   ="<<throu<<"    LUNGHEZZA MEDIA="<<lungh<<endl;
 cout<<"UTILIZZAZIONE="<<utiliz<<"    TEMPO MEDIO    ="<<tempo<<endl;
 cout<<endl;
};
```

```
int c_break(void)
{
 cout<<"Control-Break pressed."<<endl;
 return(ABORT);
};#include <iostream.h>
#include "fms.h"
#include <conio.h>
#include <dos.h>



//  FUNZIONE MAIN.

main()
{
 ctrlbrk(c_break);
 FMS rete;

 Leggi_FMS(rete);
 Tassi_arrivi(rete);

 long double * * tassi_arriv=new long double*[rete.num_tot_code + 1];
 long double * * tassi_serv=new long double*[rete.num_tot_code + 1];
 int * * buffer_coda=new int*[rete.num_tot_code + 1];
 char * * caratteri=new char*[rete.num_tot_code + 1];

 iniz_Buzacott(rete,tassi_arriv,tassi_serv,buffer_coda,caratteri);

 int posti=posti_ok(rete.num_tot_code,rete.num_utenti,buffer_coda);
 if( !posti )
```

```
 {
  clrscr();
  cout<<endl;
  cout<<"IL NUMERO DI UTENTI E' MAGGIORE DELLA SOMMA DELLE CAPACITA'."<<endl;
  delete tassi_arriv;
  delete tassi_serv;
  delete buffer_coda;
  delete caratteri;
  return 0;
 }

int capacit_ok=buffer_ok(buffer_coda,rete.num_tot_code,rete.num_utenti);
if( !capacit_ok )
 {
  clrscr();
  cout<<endl;
  cout<<"LA CAPACITA' DI UNA CODA E' MAGGIORE DEL  NUMERO DI UTENTI"<<endl;
  cout<<"NELLA RETE,IN QUESTO CASO SI DOVEVA IMMETTERE IL NUMERO DI"<<endl;
  cout<<"UTENTI AL POSTO DELL' EFFETTIVA CAPACITA'."<<endl;
  delete tassi_arriv;
  delete tassi_serv;
  delete buffer_coda;
  delete caratteri;
  return 0;
 }

clrscr();

char s;
for(int i=1; i<rete.num_tot_code;++i)
 {
  scambia_long(i,rete.num_tot_code,tassi_arriv);
```

```
   scambia_long(i,rete.num_tot_code,tassi_serv);
   scambia_int(i,rete.num_tot_code,buffer_coda);
   scambia_char(i,rete.num_tot_code,caratteri);
   Buzacott(rete,tassi_arriv,tassi_serv,buffer_coda,caratteri,i);
   cin>>s;
  };

 scambia_long(1,rete.num_tot_code,tassi_arriv);
 scambia_long(1,rete.num_tot_code,tassi_serv);
 scambia_int(1,rete.num_tot_code,buffer_coda);
 scambia_char(1,rete.num_tot_code,caratteri);
 Buzacott(rete,tassi_arriv,tassi_serv,buffer_coda,caratteri,rete.num_tot_code);

 delete tassi_arriv;
 delete tassi_serv;
 delete buffer_coda;
 delete caratteri;

 return 0;
};
```

Bibliografia

[1] Buzen J.P. (1973) **"Computational algorithms for closed queuing networks with exponential servers"**, Comm. A.C.M., vol 16, n. 9, 527-532.

[2] Eran Y. (1981) **"The study and implementation of solution for queuing networkmodels"**, Weizman Institute of Science, Dep of Appl. Math.

[3] Gordon W.J. - Newell G.F. (1967) **"Closed queuing system with exponential servers "**, Op. Res. 15, 254-265.

[4] Grimmett G. - Stirzaker **" Probability and random processes"**, Claredon Press, Oxford, 1982.

[5] Hordijk A. - Van Dijk N. (1981) **"Network of queues wiht blocking"**, Performance, K.J. Kylstra, North Holland.

[6] Jackson J.R. (1963) **"Jobshop like queueing sistem"**, Mgmt. Sci., 10, 131-142.

[7] Kelly F.P. (1979) **"Reversibility and stochastic networks"**, John Wiley and Sons, New York.

[8] Kelly F.P. (1975) **"Networks of queues with customers of different type"**, J. Appl. Prob., vol. 12, 542-554.

[9] Kelly F.P. (1976) **"Networks of queues "**, Adv. Appl. Prob., vol. 8,

416-432.

[10] Kingman J.F.C (1969) **"Markov population processes"**, J. Appl. Prob. 6, 1-18.

[11] Melamed B. (1982) **"On reversibility of queuing networks"**, Stochast. Proc. Appl. 13, 227-234.

[12] Muntz R.R. (1967) **"Poisson departure processes and queueing networks"**, IBM Research Report RC 4145 IBM Thomas J. Watson Research center, Yorktown Heights, New York.

[13] Pollet P. (1986) **"Connecting reversible Markov Processes"**, Adv. Appl. Prob. 18, 880-900.

[14] Walrand J. (1988) **"An introduction to queueing networks"**, Prentice-Hall, Englewood Cliffs, New Jersey.

[15] Yao D., Buzacott A. (1987) **"Modeling a class of Flexible Manufacturing System with reversible routing"**, Op. Res. 35, n.1, 87-93.

[16] Yao D., Buzacott A. (1986) **" Flexible manufacturing system : a rewiew of analytical models"**, Management Science, vol 32, n. 7, 890-905.

[17] Yao D., Buzacott A. **" Models of flexible manufacturing system with limited local buffer"**, Int. J. Prod. Res., vol 24, n. 1, 107-118.