

# La Simulazione

di Valter Di Dio

## Dimensionamento di un parcheggio

Dopo aver visto la scorsa volta (numero 36), le possibilità generali della simulazione, ci occupiamo, adesso, di una applicazione tra le più classiche: l'analisi di un sistema complesso in fase progettuale. Senza farsi spaventare dai paroloni, significa che proveremo a vedere come si possa controllare l'evolversi di una situazione molto complessa, e con tanti parametri su cui intervenire, utilizzando un qualsiasi personal computer e un semplice programmino in Basic.

La situazione che andremo ad analizzare è quella di un parcheggio con un ingresso e due uscite, naturalmente a pagamento, e con una disponibilità di posti, in fase progettuale, illimitata.

Quello che ci interessa sapere è innanzitutto l'affluenza prevista e quindi il numero di posti da predisporre, poi sarà interessante vedere anche il formarsi delle code in ingresso e in uscita per decidere eventualmente il numero di addetti da assegnare a ciascun casello.

Se infatti immaginiamo che la rampa di accesso permetta una fila massima di sei macchine, oltre le quali si invade la corsia di marcia della strada su cui si apre, e durante la simulazione questo numero di veicoli in attesa supera troppo spesso il valore di guardia, possiamo prevedere di mettere al casello di ingresso due addetti anziché uno in modo da dimezzare il tempo necessario alle operazioni, ad esempio prendere il tagliando di arrivo, oppure si potrà sostituire l'addetto con una macchina automatica come è stato fatto sulle autostrade.

Dovrebbe essere evidente che riducendo il tempo del servizio, a parità di arrivi, si riduce conseguentemente anche la fila di attesa.

La stessa cosa si può dire delle due uscite, che possiamo chiamare uscita Nord e uscita Sud; per queste inoltre si pone un altro problema: il modo con cui gli utenti scelgono l'una o l'altra delle uscite. Se infatti le due uscite danno sulla stessa strada non è necessario vederle, in fase di simulazione, come distinte dato che, potendo scegliere una o l'altra a piacere, è come se esistesse una sola uscita con un tempo di servizio dimezzato; ma se le due uscite danno su strade diverse gli utenti sceglieranno l'uscita in base alla loro esigenza, e dovremo quindi studiare i flussi dei probabili clienti (ricordiamo che siamo ancora in

fase di progetto) per gestire la scelta dell'uscita: più o meno è come se ciascun utente prima di uscire lanciasse un dado e decidesse poi in base al risultato. Questo comporta che le code ai due caselli di uscita sono differenti e si potrebbe verificare anche il caso in cui, durante la giornata, si modifichi la probabilità di scelta dell'uscita. In questo caso (probabilità variabile) si può provare a mettere tre addetti ai caselli di uscita di cui due fissi e uno che si sposta, a seconda dell'ora, da un casello all'altro; in termini matematici il tempo medio di sosta ai caselli di uscita varia con il trascorrere del tempo per adeguarsi ai mutati flussi di vetture.

Anche il tempo che intercorre tra un arrivo e il successivo sono tempo-dipendenti, se è infatti molto probabile che alle otto del mattino si abbia un arrivo ogni due minuti, è altrettanto probabile che dalle 11 alle 12 arrivino poche macchine. Altra variabile che si presume tempo-dipendente è la durata della sosta: chi arriva tra le sette e le nove probabilmente rimarrà fino alle 14 o più, mentre chi arriva alle 11 è facile che non si fermi più di un'ora o due. Tutti questi dati si devono in qualche modo avere prima di iniziare la simulazione. Il modo migliore di ottenere i dati in casi del genere è di munirsi di un cronometro e andare a rilevare i tempi in un parcheggio già esistente, quindi si interpolano i dati rilevati con quelli presunti (non ha senso prendere i dati di un parcheggio in periferia e utilizzarli per progettare uno al centro della città) e si utilizzeranno i risultati per la simulazione. È chiaro che tanto più precise si riveleranno le rilevazioni, tanto più verosimili saranno i risultati della simulazione. È comunque sempre possibile effettuare più simulazioni, con valori dei parametri differenti, nel caso in cui non si sia certi di alcuni dati. Può anche succedere che il risultato della simulazione non cambi sensibilmente al variare dei dati (ad esempio potrebbe non far differenza se arriva un'auto ogni tre minuti anziché ogni due) ma più spesso si scoprirà che anche piccole differenze in certe attività possono trasformare un sistema apparentemente tranquillo nel più grosso ingorgo della storia.

Quando un'attività risente particolarmente delle variazioni dei dati, quella attività si dice "critica" ed è ovviamente sulle attività critiche che si deve concentrare l'attenzione di chi progetta, perché il più piccolo errore può mettere scompiglio in tutto il sistema.

È interessante notare che molto spesso l'attività critica non è quella più evidente, e, addirittura, attività apparentemente "pacifiche" possono portare facilmente alla paralisi del sistema in caso di evento eccezionale. Capita anche spesso che due attività singolarmente non critiche diventino improvvisamente "micidiali" nel caso che si sovraccarichino tutte e due contemporaneamente; un esempio di ciò si può vedere tutti i giorni con i problemi del traffico dove piccoli incidenti e rallentamenti, magari molto lontani tra loro, concorrono in pochi minuti a bloccare completamente la circolazione.

## Il sistema

Come abbiamo già accennato, si tratta di simulare un parcheggio con una entrata, un certo numero di posti e due uscite su strade differenti.

Per quanto riguarda i dati e i parametri di lavoro, supponiamo che gli arrivi, o più esattamente la media dei tempi che intercorrono tra un arrivo e il successivo, siano variabili di ora in ora, anche il tempo medio di sosta e la relativa varianza cambiano ogni ora, mentre la probabilità di scelta dell'uscita resta costante nel corso della giornata (possiamo supporre che un'uscita porti su una grossa arteria mentre l'altra dia su una strada meno trafficata).

Lo schema del parcheggio è quello di figura 1, le righe rappresentano i flussi mentre le frecce più grosse sono le file d'attesa. Come si vede i posti macchina sono in parallelo, cioè l'utente può recarsi immediatamente al primo posto libero; spesso però non è così e prima di trovare un posto libero può essere necessario scorrere tutto il parcheggio. Nel programma, una variabile tiene conto di quanti posti occupati si sono incontrati prima di trovarne uno libero, ma nella tabella stampata questo valore (che potremmo chiamare felicità del cliente) non viene visualizzato, vedremo comunque al momento come si può accedere a questa informazione.

Per quanto riguarda le file di attesa, queste sono dimensionate per un massimo di 15 posti (aumentabile a piacere) oltre il quale il computer va in blocco mentre la segnalazione di allarme (corrispondente alla massima fila ammissibile realmente) può essere fissata molto prima.

## Le variabili casuali

Pur essendo un sistema apparentemente semplice, questo parcheggio possiede un



numero elevato di funzioni legate al caso. La prima è ovviamente l'arrivo delle auto: una volta arrivate queste devono effettuare una sosta, casuale anche questa, al casello di ingresso; poi raggiungono la piazzola di sosta e vi permangono per un periodo più o meno lungo e quindi, abbandonata la piazzola, scelgono a caso l'uscita e qui sostano, ancora casualmente, per il tempo relativo al pagamento. Solo dopo aver completato questa operazione le auto escono dal par-

dove  $P(x)$  è la probabilità di estrazione del valore  $\langle x \rangle$ . Come per la Poisson l'integrale tra meno infinito e più infinito della  $P(x)$  è uguale ad uno (questo garantisce che qualsiasi numero possa essere estratto ogni volta), ma il 95% dei valori estratti è compreso tra  $-2SG + 2SG$ , ovviamente però addensati intorno alla media.

Per quello che riguarda invece la scelta dell'uscita, la funzione che la governa si chiama Binomiale e, nella forma più este-

## Il programma

Il compito del programma Basic si può dividere in tre parti: gestire le liste di attesa, generare i numeri casuali da assegnare alle varie operazioni secondo le funzioni di distribuzione opportune e, infine, visualizzare l'andamento della simulazione.

La gestione delle liste di attesa si rivela il compito più facile.

Vediamo come si sviluppa il processo per grandi linee.

Ad ogni giro del clock (10 secondi) si determina con la Poisson se è arrivato il momento di far giungere una vettura; se si si assegna subito a questa vettura un numero (nel programma equidistribuito tra ME-VE e ME+VE) che corrisponde a quanti cicli dovrà attendere, una volta giunta al casello, prima di poter procedere. Ogni nuova vettura arrivata si aggiunge in coda al vettore della fila d'entrata  $E\%(I)$  esattamente come avviene in una fila vera; L1 indica il numero di vetture in fila e quindi il numero di elementi del vettore che sono occupati. Il vettore  $E\%(I)$  contiene quindi L1 numeri che corrispondono ai rispettivi tempi di attesa delle L1 vetture.

Ogni giro di clock (corrispondente ad un ciclo del programma) se L1 è maggiore di zero, e cioè se ci sono macchine in fila, si decrementa di uno il valore del primo elemento del vettore e, quando questo è arrivato a zero, la macchina lascia il casello e la fila compie un vero e proprio "passo avanti". Alla vettura che ha lasciato il casello viene assegnato ora il valore del tempo di sosta (distribuito normalmente con media MU e varianza SG) e, trovato un posto vuoto (valore -1) nella lista del parcheggio PD%(I), vi viene depositato appunto quel valore. Ad ogni giro del programma tutti i numeri maggiori di zero contenuti nel vettore vengono decrementati di uno (il tempo di sosta passa per tutti) e se una o più vetture arrivano a zero, queste vengono inviate alle uscite. Anche qui, dopo aver scelto quale uscita (vedremo poi come), si assegna un tempo di permanenza al casello e si mette in coda la vettura nei vettori U 1% o U 2% che sono gestiti nel loop principale esattamente come l'entrata.

## Generazione di variabili casuali

Il problema che si pone con i personal computer è la mancanza di generatori casuali con distribuzione qualsiasi (o almeno con le più comuni). Per ottenerle occorre lavorare, non poco, con il generatore interno e alcune funzioni di conversione che devono essere nello stesso tempo precise e veloci. Il metodo usato la volta scorsa (simulazione di sala gruppi) per verificare se un numero RND generato uniformemente poteva appartenere o no ad una certa distribuzione si può usare solo quando questo calcolo viene eseguito un numero relativamente piccolo di volte all'interno del programma, altrimenti si rischia di ottenere una simulazione "in tempo reale" o peggio rallentata.

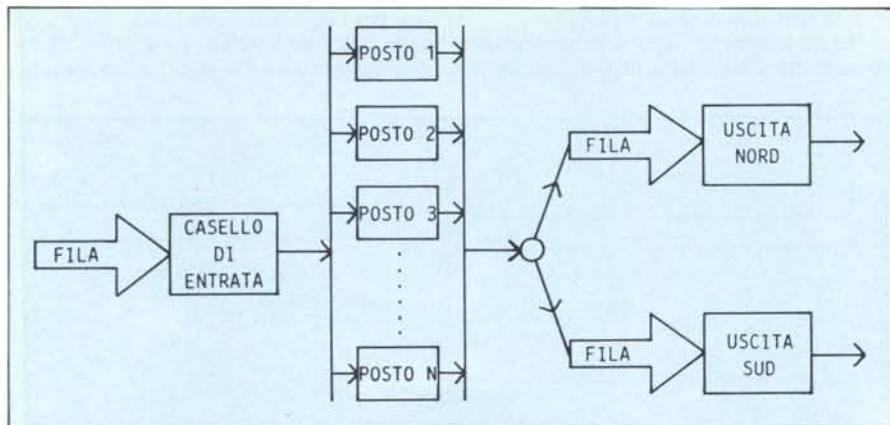


Figura 1 - Diagramma di flusso del parcheggio. Le frecce grandi rappresentano le code ai caselli di entrata e di uscita, quelle piccole i sensi di marcia.

cheggio e quindi dal sistema (i più pignoli possono ancora seguire le auto nella successiva fila d'attesa necessaria ad immettersi nel traffico!).

Tutte le volte che abbiamo detto caso non si intendeva semplicemente un numero RND (I) generato dal computer, ma un numero estratto si a caso ma con una precisa distribuzione, una certa media e una data varianza.

La distribuzione che gestisce il succedere di eventi in una fila d'attesa classica è la Poisson:

$$P(t) = \sum_{i=0}^t \frac{e^{-a} a^i}{i!}$$

che, tradotta in parole, rappresenta la probabilità che si verifichi un arrivo dopo che sono trascorse t unità di tempo dal precedente e sapendo che l'intervallo medio fra gli arrivi è  $\langle a \rangle$ . La varianza della Poisson è uguale alla media e quindi ad  $\langle a \rangle$ . Il simbolo  $i!$  (si legge i fattoriale) corrisponde al prodotto dei primi  $\langle i \rangle$  numeri naturali: quindi  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ .

Per t che tende a infinito la sommatoria precedente tende a uno e quindi all'evento certo.

Per quanto riguarda invece il tempo del servizio ai caselli e il tempo di sosta delle vetture questo si distribuisce "Normalmente"; detta MU la media e SG la varianza, la funzione di distribuzione è la seguente:

$$P(x) = \frac{1}{\sqrt{2\pi SG}} \cdot e^{-\frac{(x-MU)^2}{2SG}}$$

sa, corrisponde a quanto segue:

$$P(A) = \frac{n!}{(n-k)!k!} \cdot p^k \cdot (1-p)^{n-k}$$

Qui il primo fattore corrisponde al numero di combinazioni tra n elementi presi a k a k, p è la probabilità che l'evento A si verifichi e 1-p è ovviamente la probabilità opposta. P(A) è allora la probabilità che esca K volte su N il risultato A. Facciamo una prova.

Qual è la probabilità che escano due sei lanciando cinque dadi? La probabilità  $\langle p \rangle$  che esca il sei su un lancio è ovviamente  $1/6$  (.1666 periodico) 1-p diventa quindi .8333 per cui sostituendo i numeri alla formula si ottiene:

$$P(6) = \frac{5!}{(5-2)!2!} \cdot 0.167^2 \cdot 0.833^{5-2}$$

che è uguale a 0.1606; quindi di poco inferiore a quella di fare 6 su un lancio secco!

Nel nostro caso però le cose si semplificano molto perché il "lancio" è uno solo, tutto il primo fattore diventa perciò uguale ad uno ad essendo  $(n-k) = 0$  scompare anche l'ultimo fattore, per cui la probabilità di scelta dell'uscita 1 diventa:

P (uscita 1) = p  
e l'evento opposto (uscita 2)

P (uscita 2) = 1-p

Era del resto intuitivo che se gli utenti scelgono di uscire da  $\langle 1 \rangle$  con probabilità .6(60%) usciranno da  $\langle 2 \rangle$  il rimanente 40%.



Abbiamo perciò usato quel metodo solo per gli arrivi (comunque ottimizzandolo) che sono eventi relativamente rari. Per la normale si è usata la classica conversione Polare proposta (e dimostrata) da Knuth nel suo *The Art Of Programming* più noto come la "Bibbia di Knuth". Per il tempo di permanenza ai caselli ci siamo accontentati dell'equidistribuita per non appesantire troppo il programma e soprattutto per non aumentare oltre i già lunghi tempi di elaborazione.

**Da RND(1) a Poisson**

Il punto critico della distribuzione di Poisson è il calcolo del fattoriale; primo per il tempo che occorre, secondo perchè il computer (Apple II) non calcola fattoriali oltre 33!

Per "fortuna" anche il valore  $a^i$  (vedi formula) sballa dopo  $20 \cdot 30$  per cui la Poisson calcolabile su un personal non può avere tempo medio di attesa superiore a 20; è comunque vero che il valore di  $t$  necessario ad avvicinarsi ad 1 quel tanto che basta a rendere l'arrivo pressochè certo è molto inferiore a 20, per cui è sufficiente calcolare i fattoriali fino a 25.

Il calcolo dei fattoriali è stato fatto preventivamente e tabellato nel vettore F (I); per cui  $F(5) = 120$ ,  $F(8) = 40320$  e così via. Anche tutti i valori della Poisson per  $t$  che va da 0 a 25 sono stati preventivamente calcolati e messi in un vettore; visto poi che di distribuzioni di Poisson ne servivano 14 (una per ciascuna ora, con medie diverse), si è messo il tutto in una "matriciona"  $PA\%(I,J)$  dove I sono le ore e J le decine di secondi trascorsi dall'ultimo arrivo.

All'interno del programma si calcola un numero RND(1) e si vede se è minore o maggiore di  $PA\%(I,J)$ ; se il numero generato è minore del corrispondente Poissoniano si verifica un arrivo, altrimenti no. Calcolate che con media 6 la Poisson vale .9999 già dopo 8 cicli.

La parte di programma che genera la Matrice va dalla riga 1200 alla riga 1280. La matrice viene usata alle righe 370 - 390.

**Da RND(1) a Normale**

Il metodo di conversione utilizzato per generare un numero con distribuzione normale da una coppia di numeri equidistribuiti è particolarmente semplice e quindi veloce, la precisione è più che sufficiente per gli usi comuni ma molto dipende dalle caratteristiche del generatore interno della macchina. In effetti il generatore RND(1) di un computer non genera realmente numeri casuali (solo un'urna lo può fare) ma con speciali algoritmi basati sul modulo di divisioni e troncamenti vari permette di ottenere, partendo da un numero base detto seme, un secondo numero apparentemente indipendente dal primo, usando questo come seme si ottiene il successivo e così via.

È chiaro perciò che partendo dallo stesso seme si otterrà sempre la stessa serie.

Un secondo difetto è il fatto che queste serie, essendo il risultato di operazioni di modulo, sono per forza cicliche, dopo un certo numero di estrazioni riatterremo così il seme iniziale e quindi di nuovo la stessa serie.

Utilizzando dei semi molto grandi e delle divisioni per numeri particolari si garantisce che la serie sia abbastanza lunga da non creare problemi ad un uso comune.

La media di una variabile equidistribuita tra A e B deve essere ovviamente  $A + B/2$  e la varianza sarà  $(B-A)^2/12$ .

Per conoscere l'efficienza del generatore di numeri casuali della propria macchina

basta perciò generare un numero elevato di valori e calcolarne:

$$\text{Media } \sum_{i=1}^n X_i$$

$$\text{e varianza } \frac{\sum_{i=1}^n (X_i - M)^2}{n}$$

e vedere se corrispondono ai valori calcolati per l'equidistribuita (vedi figura 3). Se la media tende ad essere inferiore si dice che il generatore è rosa se invece prevalgo-

<p>ORE 7:18:20</p> <p>*****</p> <p>ATTIVITA' - ATTESA - FILA MAX - FAULT</p> <p>INGRESSO 0 1 0</p> <p>USCITA 1 0 0 0</p> <p>USCITA 2 0 0 0</p> <p>*****</p> <p>POSTI: PREVISTI - OCCUPATI - MASSIMO</p> <p>150 150 150</p> <p>*** ESAURITO ***</p>				<p>USCITA 2 0 0 0</p> <p>*****</p> <p>POSTI: PREVISTI - OCCUPATI - MASSIMO</p> <p>150 150 150</p> <p>*** ESAURITO ***</p>			
<p>ORE 8:20:10</p> <p>*****</p> <p>ATTIVITA' - ATTESA - FILA MAX - FAULT</p> <p>INGRESSO 1 1 0</p> <p>USCITA 1 0 0 0</p> <p>USCITA 2 0 0 0</p> <p>*****</p> <p>POSTI: PREVISTI - OCCUPATI - MASSIMO</p> <p>150 18 18</p>				<p>ORE 19:28:20</p> <p>*****</p> <p>ATTIVITA' - ATTESA - FILA MAX - FAULT</p> <p>INGRESSO 0 5 0</p> <p>USCITA 1 8 8 2</p> <p>USCITA 2 0 4 0</p> <p>*****</p> <p>POSTI: PREVISTI - OCCUPATI - MASSIMO</p> <p>150 103 150</p>			
<p>ORE 11:46:30</p> <p>*****</p> <p>ATTIVITA' - ATTESA - FILA MAX - FAULT</p> <p>INGRESSO 4 5 3</p> <p>USCITA 1 0 7 1</p> <p>USCITA 2 0 9 2</p> <p>*****</p> <p>POSTI: PREVISTI - OCCUPATI - MASSIMO</p> <p>150 92 92</p>				<p>ORE 19:35:30</p> <p>*****</p> <p>ATTIVITA' - ATTESA - FILA MAX - FAULT</p> <p>INGRESSO 0 5 0</p> <p>USCITA 1 8 8 4</p> <p>USCITA 2 0 4 0</p> <p>*****</p> <p>POSTI: PREVISTI - OCCUPATI - MASSIMO</p> <p>150 66 75</p>			
<p>ORE 8:46:10</p> <p>*****</p> <p>ATTIVITA' - ATTESA - FILA MAX - FAULT</p> <p>INGRESSO 0 5 0</p> <p>USCITA 1 0 0 0</p> <p>*****</p> <p>POSTI: PREVISTI - OCCUPATI - MASSIMO</p> <p>150 63 75</p>				<p>ORE 19:37:00</p> <p>*****</p> <p>ATTIVITA' - ATTESA - FILA MAX - FAULT</p> <p>INGRESSO 1 5 0</p> <p>USCITA 1 1 2 4</p> <p>USCITA 2 0 4 0</p> <p>*****</p> <p>POSTI: PREVISTI - OCCUPATI - MASSIMO</p> <p>150 63 75</p>			

Figura 2 - Alcune stampe di uscita della simulazione in vari momenti della giornata. La colonna FAULT indica quante volte una macchina in uscita ha trovato una coda superiore a quella di guardia (nell'esempio era di sei auto in entrata e di cinque in uscita). La scritta ESAURITO, che compare in basso, corrisponde ad un semaforo posto all'ingresso del parcheggio che blocca l'accesso quando non ci sono più posti a disposizione.



no i valori alti (media superiore) il generatore sarà azzurro.

Se la varianza differisce di molto, la distribuzione, molto probabilmente, non sarà uniforme e serviranno altre prove per scoprire di quale distribuzione si tratta (una prima prova potrebbe essere l'indice di simmetria: tanti valori sotto la media quanti sopra).

I generatori di variabili casuali della Microsoft e quindi Apple, Commodore e MSX, sono abbastanza precisi ma leggermente rosa (di pochissimo) e perciò tranquillamente utilizzabili, per altre macchine occorre provare.

Tornando alla Normale, se il generatore è buono il metodo polare garantisce buoni risultati in tempi di elaborazione accettabili.

La conversione funziona in tre stadi e, partendo da due numeri RND equidistribuiti tra 0 e 1 (incluso) genera una coppia di numeri casuali appartenenti ad una distribuzione Normale Standardizzata: cioè con media 0 e varianza 1.

Si comincia generando due numeri casuali U1 ed U2, poi si calcola:

$$V1 = 2 \times U1 - 1 \text{ (perciò } -1 \leq V1 \leq +1 \text{)}$$

$$V2 = 2 \times U2 - 1 \text{ (idem)}$$

poi si calcola

$$S = V1^2 \times V2^2$$

se S è maggiore o uguale ad 1 si scarta tutto e si ricomincia, altrimenti si calcola

$$N1 = V1 \times \text{SQR}((-2 \times \text{LOG}(S))/S)$$

$$N2 = V2 \times \text{SQR}((-2 \times \text{LOG}(S))/S)$$

N1 ed N2 sono distribuiti normalmente con media zero e varianza uno. A questo punto basta moltiplicarli per la varianza voluta e sommarci la media per avere i valori definitivi. Il calcolo della normale viene eseguito dalla riga 900 alla riga 940, naturalmente si calcola un solo valore dato che il secondo non serve.

```

100 A$ = "*****"
110 :
120 NP = 150: REM NUMERO DI POSTI
130 U1 = .4:U2 = 1 - U1: REM SCELTA DELL'USCITA
140 EM = 15:UM = 15: REM FILA MAX
150 EF = 6:UF = 5: REM ALLARME DI CODA
160 ME = 3:VE = 2: REM STATISTICHE DI ENTRATA
170 M1 = 4:V1 = 3: REM STATISTICHE USCITA 1
180 M2 = 3:V2 = 4: REM STATISTICHE USCITA 2
190 E = EXP (1):TH = 7
200 DIM PAZ(15,25),PDZ(NP),E%(EM),U1%(UM),U2%(UM),F(25)
210 :
220 TEXT : HOME
230 PRINT : PRINT "INIZIO SIMULAZIONE"
240 PRINT : PRINT " APERTURA ORE 7:00"
250 PRINT : PRINT " CHIUSURA ORE 21:00"
260 PRINT : PRINT "CARICAMENTO MATRICI"
270 GOSUB 1200
280 :
300 REM *** INIZIO CICLO ***
310 READ MU,SG
320 TD = TD + 1: IF TD = 6 THEN TD = 0:TH = TH + 1
330 IF TH = 60 THEN TH = 0:TH = TH + 1: READ MU,SG
340 GOSUB 2010
350 IF TH = 21 THEN END
360 REM *** ARRIVI ***
365 IF EE THEN 470
370 X = PAZ(TH - 7,R):R = R + 1
380 A = RND (1) * 10000
390 IF (X < A) THEN 470
400 R = 0:L1 = L1 + 1
410 E%(L1) = RND (1) * VE + ME - VE / 2
420 IF L1 > EX THEN EX = L1
430 IF L1 > EF THEN F1 = F1 + 1
460 REM *** SERVE LISTA DI ENTRATA ***
470 IF L1 = 0 THEN 530
480 E%(0) = E%(0) - 1
490 IF E%(0) > 0 THEN 530
500 FOR I = 0 TO L1 - 1:E%(I) = E%(I + 1): NEXT
510 GOSUB 810
520 REM *** SERVE LISTA DI PARCHEGGIO ***
530 FOR I = 0 TO PH
540 IF PDZ(I) < 0 THEN 570
550 IF PDZ(I) = 0 THEN GOSUB 1010
560 PDZ(I) = PDZ(I) - 1
570 NEXT I
580 REM *** SERVE LISTA DI USCITA 1 ***
590 IF L2 = 0 THEN 650
600 U1%(0) = U1%(0) - 1
610 IF U1%(0) > 0 THEN 650
620 FOR I = 0 TO L2 - 1:U1%(I) = U1%(I + 1): NEXT
630 L2 = L2 - 1
640 REM *** SERVE LISTA DI USCITA 2 ***
650 IF L3 = 0 THEN 320
660 U2%(0) = U2%(0) - 1
670 IF U2%(0) > 0 THEN 320
680 FOR I = 1 TO L3 - 1:U2%(I) = U2%(I + 1): NEXT
690 L3 = L3 - 1: GOTO 320
800 REM *** MACCHINA IN ENTRATA ***
810 L1 = L1 - 1:PO = PO + 1: IF PO > PX THEN PX = PO
820 IF PO = NP - L1 THEN EE = 1
830 FOR J = 0 TO PH
840 IF PDZ(J) < 0 THEN 900
850 NEXT
860 PH = PH + 1
870 GOTO 830
900 REM *** CALCOLO TEMPO DI SOSTA ***
910 G1 = RND (1) * 2 - 1:G2 = RND (1) * 2 - 1
920 S = G1 * G1 + G2 * G2: IF S > 1 THEN 910
930 XX = G1 * SQR ((-2 * LOG (S)) / S)
940 PDZ(J) = XX * SG + MU
950 RETURN

1000 REM *** MACCHINA IN USCITA ***
1010 PO = PO - 1: IF I < PH THEN 1020
1015 PH = PH - 1: IF PDZ(PH) < 0 THEN 1015
1020 A = RND (1):EE = 0
1030 IF A > U2 THEN 1120
1040 REM *** ESCE DAL CASELLO 1 ***
1050 L2 = L2 + 1
1060 IF L2 > UF THEN F2 = F2 + 1
1070 U1%(L2) = RND (1) * V1 + M1 - V1 / 2
1080 IF L2 > UX THEN UX = L2
1100 RETURN
1110 REM *** ESCE DAL CASELLO 2 ***
1120 L3 = L3 + 1
1130 IF L3 > UF THEN F3 = F3 + 1
1140 U2%(L3) = RND (1) * V2 + M2 - V2 / 2
1150 IF L3 > UY THEN UY = L3
1160 RETURN
1170 :
1200 F(0) = 1
1210 FOR I = 1 TO 25:F(I) = F(I - 1) * I
1220 NEXT
1230 FOR I = 0 TO 14: READ A
1235 PAZ(I,0) = (E ^ - A) * 10000
1240 FOR J = 1 TO 25
1250 PA = (E ^ - A) * (A ^ J) / F(J)
1260 PAZ(I,J) = PAZ(I,J - 1) + PA * 10000
1270 NEXT : VTAB 12: HTAB 1: PRINT " RIGA: "I
1280 NEXT
1290 FOR I = 0 TO NP:PDZ(I) = - 1: NEXT
1300 RETURN
1310 :
2000 REM *** S T A M P A ***
2010 IF PEEK (- 16384) = 208 THEN GET X$:PR# 1
2020 HOME : PRINT SFC(10):"ORE "TH:"TM":"TD"O " : PRINT
2030 PRINT : PRINT A$
2040 PRINT
2050 PRINT "ATTIVITA' - ATTESA - FILA MAX - FAULT"
2060 PRINT
2070 PRINT "INGRESSO": TAB(15):L1: TAB(25):EX: TAB(35):F1
2080 PRINT
2090 PRINT "USCITA 1": TAB(15):L2: TAB(25):UX: TAB(35):F2
2100 PRINT
2110 PRINT "USCITA 2": TAB(15):L3: TAB(25):UY: TAB(35):F3
2120 PRINT
2130 PRINT A$
2140 PRINT
2150 PRINT "POSTI: PREVISTI - OCCUPATI - MASSIMO"
2160 PRINT
2170 PRINT TAB(11):NP: TAB(22):PO: TAB(33):PX
2180 PRINT : IF EE THEN PRINT "*** ESAURITO ***": GOTO 2200
2190 PRINT
2200 PR# 0: RETURN
2210 :
10000 REM DATI ORARI
10001 DATA 6,8,10,15,15,10,6,15,15,8,6,6,8,10,20
10002 :
10007 DATA 2520,1080
10008 DATA 2160,1080
10009 DATA 2880,360
10010 DATA 1080,720
10011 DATA 1080,720
10012 DATA 720,360
10013 DATA 1800,720
10014 DATA 1800,360
10015 DATA 1440,360
10016 DATA 1080,360
10017 DATA 720,360
10018 DATA 360,180
10019 DATA 360,180
10020 DATA 180,90
10021 DATA 180,90
    
```

Listato 1 - Programma Basic (Apple) per simulare una giornata di funzionamento di un parcheggio con un casello di entrata e due di uscita, il numero di posti, così come i tempi medi di permanenza e di servizio, possono essere cambiati a piacere sostituendo i valori delle variabili che si trovano nelle righe comprese tra la 120 e la 180.



```

10 FOR I = 1 TO 10000      JRUN
20 X = RND (1)
22 X1 = X1 + X           MOMENTO 1 (.499) .502875298
23 X2 = X2 + X * X      MOMENTO 2 (.333) .336213781
24 X3 = X3 + X * X * X  MOMENTO 3 (.25 ) .252536229
25 X4 = X4 + X ^ 4      MOMENTO 4 (.2 ) .202202786
30 NEXT
40 I = I - 1
50 M1 = X1 / I
52 M2 = X2 / I
53 M3 = X3 / I
54 M4 = X4 / I
60 PRINT "MOMENTO 1 (.499) "M1
61 PRINT "MOMENTO 2 (.333) "M2
62 PRINT "MOMENTO 3 (.25 ) "M3
63 PRINT "MOMENTO 4 (.2 ) "M4

```

Figura 3 - Programma per il controllo della precisione del generatore di numeri casuali. I risultati devono essere il più vicino possibile al valore teorico scritto tra parentesi. Come si vede, il generatore dell'Apple è praticamente perfetto.

## Da RND(1) a Binomiale

È questo il caso più semplice in quanto basta controllare se il numero generato, che è già tra zero e uno, è minore o maggiore della probabilità dell'evento da scegliere. In pratica se dobbiamo scegliere A nel 25% dei casi, B nel 45% dei casi e C nel rimanente 30% delle volte, si genera un numero RND(1) e se è minore di .25 si sceglie A, se è compreso tra .26 e .65 si sceglie B, se è invece maggiore di .7 si deciderà per C.

La Binomiale viene usata nel programma per la scelta dell'uscita alle righe 1020 e 1030.

## Note al programma

Le prime righe del programma contengono i parametri statistici dei vari caselli e i limiti delle code; in fondo, alla riga 10000, si trovano i dati relativi agli arrivi medi ora per ora e dalla 10007 in poi la media e la varianza del tempo di sosta per chi arriva ad una certa ora (l'ora è uguale alle ultime cifre del numero di riga); ricordarsi che il 95% dei valori generati con distribuzione normale è compreso tra  $-2 \times SG$  e  $+2 \times SG$ , quindi con media 2000 e varianza 500 si ottengono valori compresi tra 1000 e 3000 salvo casi rari. Per convertire le ore in decine di secondi, che è l'unità di misura del programma, bisogna ovviamente moltiplicare per 360.

La gestione delle code, come spiegato precedentemente, non è la migliore possibile (è comunque la più facile da spiegare); si potrebbe risparmiare lo scorrimento in avanti della fila creando un puntatore al "primo della fila" e muovendo lui anziché la coda: questo rende la velocità costante anche per code molto lunghe.

Per quanto riguarda la gestione delle macchine in sosta, il ciclo FOR...NEXT

della riga 530 va da zero ad un fantomatico PH; questo è il puntatore alla posizione più alta (High) occupata dentro il vettore PD% (posti disponibili) e in questo modo la velocità di esecuzione di un loop non dipende più dalla dimensione del parcheggio, ma solo dal numero di auto in sosta. Il valore di PH sale ogni volta che una macchina in arrivo non trova posti liberi tra i primi PH (riga 860), e scende fino al primo posto occupato che trova, quando la macchina in posizione PH esce (righe 1010,1015). Per liste brevi, fino a cinquanta posti, questo lavoro non è necessario e i cicli possono andare tranquillamente da zero al massimo numero di posti previsti.


Con 50 macchine dentro un ciclo dura 8 decimi di secondo mentre, con 150, sale a 3 secondi, per diventare 4 secondi quando arriva una macchina che deve cercare un posto. La Dimension massima accettata dall'Apple per il numero di posti è di 15000, ma la simulazione diventerebbe certamente troppo lenta.

Come accennato precedentemente, è possibile sapere quanti posti occupati ha incontrato una vettura in ingresso prima di trovarne uno libero. La variabile è la J del ciclo FOR...NEXT di riga 830. Per ottenere in uscita questo valore basta aggiungere la seguente riga:

```
2195 PRINT "Felicita" = "J
```

## Conclusioni

Con questo esempio abbiamo terminato la gestione delle liste di attesa e anche buona parte della trattazione matematica; a questo punto molti saranno in grado di applicare quanto detto anche ad altri problemi: gestione delle scorte in un magazzino o delle telefonate ad un centralino e, perché no, dell'arrivo e invio di messaggi in una rete di computer.

Appuntamento alla prossima volta per una nuova, divertente applicazione. 

## Dove trovare le stampanti telcom

### Agenti

**BARI:** L & L COMPUTER - Largo 2 Giugno 4 tel. 080/224.277  
**FIRENZE:** MARCO BARONI - Via Malbran 51 tel. 055/350.471  
**GENOVA SAMPIERDARENA:** ARGE snc Via N. Ronco 2 canc. - tel. 010/41.38.04  
**NAPOLI:** E.D.L. srl - Via Coriolano 3/D tel. 081/63.23.35  
**PADOVA (TENCAROLA SELVAZZANO):** SITAL srl Via Euganea 7 - tel. 049/63.75.61  
**ROMA:** INFO - Via Flavio Domiziano 10 tel. 06/5126700 - 5138023 - 5127381

### Punti di vendita autorizzati:

**ANCONA (FALCONARA):** PAMO COMPUTERS SHOP snc - Via Leopardi 12 - tel. 071/917.35.20  
**BARI:** TECHNCOMP srl - Via Villari 6 tel. 080/21.68.68  
**BOLOGNA:** INFORMATICA srl - Via Mascarella 116 tel. 051/24.03.69  
**BOLZANO:** DATAPLAN - Via Cassa di Risparmio 9 tel. 0471/47.721  
**CARMAGNOLA (TO):** ESSETIERRE snc di F.lli MICELI - Via Valobra 186 - tel. 011/977.18.93  
**CHIAVARI:** E.L.C.O. snc - Via R. Orsi 44 tel. 0185/32.13.47  
**CIVITANOVA MARCHE:** ELETTROCASA V.le F.lli Matteotti 63/71 - tel. 0733/73.621  
**FERRARA:** C. & P. srl - Via Corte Vecchia 67 tel. 0532/48.648  
**GAETA:** BIT COMPUTERS - Lungomare Caboto 74 tel. 0771/47.01.68  
**GENOVA:** RAPPRE-EL - Via Borgoratti 23/r tel. 010/31.68.88-36.35.72  
**LATINA:** BIT COMPUTERS - Via A. Diaz 14 tel. 0773/49.52.85  
**MERANO:** COMPUTER MARKET Via S. Maria del Conforto - tel. 0473/36.133  
**MESSINA:** SOFIN spa - Via Don Blasco 75 tel. 090/29.23.987  
**MILANO:** ALL'INFORMATICA SHOP - Via Lazzaretto 2 - tel. 02/28.70.105 • BITOMANIA snc - Via Rosolino Pilo 3 - tel. 02/20.43.130 • L'UFFICIO 2000 - Via Ripamonti 213 - tel. 02/56.96.570  
**MONZA:** MICROSHOP - Via Corte Longa 15/17 tel. 039/38.98.50  
**PARMA:** BIT SHOW - Borgo Parente 14/E tel. 0521/25.014  
**PERUGIA:** MICROCOGIT sas - Via del Filosofi 24 tel. 075/35.132  
**REGGIO CALABRIA:** SOFIN spa - Via S. Francesco di Paola 108/d - tel. 0965/25.103  
**REGGIO EMILIA:** COMPUTER CENTER srl Via Dante Alighieri 4 - tel. 0522/31.965  
**ROMA:** ALTEC srl - Via Monte D'Onorio 35 - tel. 06/79.41.755 • BIT COMPUTERS - Via Flavio Domiziano 10 - tel. 06/51.26.700 • 51.38.023 - 51.27.381 • Via Francesco Satali 55, 57, 59 - tel. 06/63.86.096 - 63.86.146 • Via Tuscolana 350/350a - tel. 06/79.439.80 • Via Nemorense 14/16 - tel. 06/85.82.96 • Viale Junio 333/335 - tel. 06/81.70.632 • COMPUTER MARKET srl - P.za S. Donà di Piave 14 - tel. 06/79.45.43 • DATA OFFICE spa - Via Sicilia 205 - tel. 06/47.54.568 • EPTA srl - Via Verona 30 - tel. 06/42.84.13 • MEMORY COMPUTER srl - Via Aureliana 39/43 - tel. 06/47.58.366 • PERSONAL COMPUTER - P.za Pio XI, 26 - tel. 06/63.80.353  
**SAVONA:** GANORA INFORMATICA - Via Torino 59 tel. 019/36.204  
**TORINO:** INPUT COMPUTER STUDIO sas - C.so Einaudi 8 - tel. 011/59.55.94 • SOFTEC COMPUTER srl - Via Juvvra 24 - tel. 011/83.96.446  
**TRIESTE:** COMPUTER MARKET srl Via Val di Rivo 6 - tel. 040/61.946  
**VERONA:** ARMUFFICIO snc - Via Guglielmo Marconi 36/38 - tel. 045 / 33.812-38.874 • GUIDO BIANCHI & C. - Via Saffi 1 - tel. 045/59.00.11  
**VITERBO:** BIT COMPUTERS - Via Palmanova 12c tel. 0761/22.39.77



ANCHE PER MICRO  
E PERSONAL

# FIORISCONO LE MARGHERITE

CON **JUKI**  
NATURALMENTE

JUKI serie 6000

Stampanti a margherita.

10 - 18 - 40 caratteri al secondo.

110 - 132 - 158 - 197 colonne (a 10, 12, 15 car/pollice).

Ampia scelta di margherite.

Fogli singoli - moduli continui.

Interfacce standard, speciali, bufferizzate.

Nessun problema di collegamento: le margherite JUKI "crescono bene"  
nel campo degli home, personal, micro e minicomputer.



gioca la carta  
**telcom**

Telcom s.r.l. - 20148 Milano - Via M. Civitali, 75  
Tel. 4047648 (3linee ric. aut.) - Telex 335654 TELCOM I