

ESERCIZI DI SIMULAZIONE DISCRETA 1

1. **(File d'attesa: modello base)** Modellare una fila d'attesa e osservare l'andamento della lunghezza della coda per un periodo di 12 ore, considerando diverse opzioni:
- tempo di interarrivo: 1 ora, 30 minuti, esponenziale di media 15 minuti, gamma di media 2 ore e deviazione standard 30 minuti;
 - tempo di servizio: 1 ora, triangolare min 30-modà 75-max 90 minuti;
 - la capacità della coda è pari a 5 e i clienti in eccesso arrestano la simulazione oppure lasciano il sistema;
 - gli utenti lasciano la coda se aspettano più di due ore senza essere serviti;
 - uno o due serventi identici.

Visualizzare un grafico che riporti l'andamento della lunghezza della coda nel tempo e dell'evoluzione della lunghezza media della coda nel tempo.

2. **(Parametri di coda)** Considerare una fila d'attesa di cui si descrivono il processo di arrivo degli utenti e il processo di servizio. Ricavare per via simulativa (e analitica dove possibile)

- a. parametri di coda,
- b. utilizzo dei serventi,
- c. tempo nel sistema,

considerando i seguenti parametri di arrivo e servizio:

- arrivi a frequenza costante pari a due arrivi al minuto e tempo di interarrivo costante, un servente, tempo di servizio costante pari a 30 secondi;
- arrivi a frequenza costante pari a due arrivi al minuto e tempo di interarrivo costante, due serventi, tempo di servizio costante pari a 45 secondi;
- arrivi a frequenza costante pari a un arrivo al minuto e tempo di interarrivo costante, due serventi, tempo di servizio distribuito secondo una distribuzione rettangolare con tempo massimo pari a 90 secondi e minimo pari a 30 secondi;
- arrivi a frequenza costante pari a un arrivo al minuto e tempo di interarrivo costante, tre serventi, tempo di servizio distribuito secondo una distribuzione gamma con tempo medio pari a 120 secondi e deviazione standard pari a 30 secondi;
- arrivi a frequenza costante pari a un arrivo al minuto e tempo di interarrivo costante, tre serventi, tempo di servizio distribuito secondo una distribuzione normale con tempo medio pari a 120 secondi e deviazione standard pari a 30 secondi.

Ripetere i casi sopra elencati variando solo gli arrivi con arrivi casuali e indipendenti con la stessa media.

3. **(Serie di code)** Considerare due file d'attesa in serie, con arrivi casuali e indipendenti con frequenza media di un arrivo al minuto, primo servizio con due serventi e tempo di servizio normale con media pari a 110 secondi e deviazione standard pari a 20 secondi, secondo servizio con sei serventi con tempo distribuito secondo una gamma con media 5 minuti e deviazione standard di 1 minuto.

Si vogliono simulare 500 arrivi e individuare eventuali bottleneck del sistema.

Ripetere l'analisi nell'ipotesi di dotare il 20% degli arrivi di una speciale autorizzazione che permette di evitare il passaggio per la prima fila della serie.

Suggerimenti per la simulazione con Anylogic

1. Utilizzare i blocchi *source*, *queue*, *delay* e *sink* della *Process Modeling Library* (PML): si suggerisce una lettura della loro descrizione (*Help – Anylogic help – Library Reference Guides – Process Modeling Library – Process Modeling Library Blocks – Source/Sink/Delay/Queue*).

Impostare l'unità di misura del tempo (ore in questo caso) tra le proprietà generali del modello (Model Time Units) e fare attenzione alle unità di misura associate ai parametri dei blocchi (ad esempio tempi di interarrivo).

- Tempi di interarrivo: utilizzare *Arrivals defined by ... Interarrival times* (*rate* può essere usato solo se i tempi di interarrivo sono distribuiti secondo un'esponenziale di media $1/\text{rate}$).
- Distribuzioni: *Help – Anylogic help – Anylogic help – Stochastic Modeling – Probability Distributions* (usate *exponential* e *triangular*).
- Capacità: impostare la *capacity* del blocco *queue*. La simulazione si arresta automaticamente in caso di eccesso con un errore, perché tutte le entità passanti devono essere raccolte in un blocco *sink*. Per evitare l'arresto della simulazione in caso di eccesso, impostare *enable preemption* e raccogliere le entità respinte in uscita dalla porta in alto a sinistra.
- Massimo tempo di attesa: impostare *Enable exit on timeout* e *timeout*, e raccogliere le entità in uscita dalla porta in alto a destra.
- Numero di serventi: impostare la *capacity* del blocco *delay*.

Per la visualizzazione dei risultati, usiamo componenti della palette *Analysis*, in questo caso dei *Time Plot* (vedere l'*Help* per il loro utilizzo).

Durante la simulazione, facendo click su ciascuna entità residente si apre una *probe* che riporta dei dati sullo stato dell'entità stessa.

2. Alcune statistiche possono essere raccolte automaticamente abilitando l'attributo *Force Statistics Collection* presente in molti blocchi della PML. Per l'esercizio sono usate:

- `queue.statsSize.mean()` (lunghezza media di una coda)
- `delay.statsUtilization.mean()` (tasso di utilizzo medio di un servente)

Per raccogliere statistiche sui tempi di attraversamento delle entità passanti si possono utilizzare delle coppie di blocchi *TimeMeasureStart* e *TimeMeasureEnd* (vedi *Help* di AnyLogic su questi blocchi e *Help – FAQ – How to ... - How can I measure agent's "time in system" or "length of stay"?*). Notare che il tempo medio di attraversamento tra i punti di ingresso e di uscita di una coppia di questi blocchi è disponibile tramite

`nomeBloccoTimeMeasureEnd.distribution.mean()`.

Distribuzione rettangolare ottenuta tramite la funzione *uniform*. La distribuzione gamma richiesta è ottenuta con `gamma(16,0.125) ... minutes`. La distribuzione normale richiesta è ottenuta con `normal(0.5,2) ... minutes` (notare che il primo parametro è sigma, il secondo la media).

Arrivi (in generale eventi) "casuali e indipendenti" sono modellati con una distribuzione esponenziale.

3. Per simulare un numero fissato di arrivi, usare l'attributo *Limited number of arrivals* del blocco *source*.

Per valutare la presenza di bottleneck, plottiamo la lunghezza media (e massima) delle due code, che possiamo ottenere abilitando, come visto, l'attributo *Force Statistics Collection*, e utilizzando un *Bar Chart* dalla palette *Analysis*.

Ribadiamo la necessità di ripetere la simulazione un numero congruo di volte per poter avere una statistica significativa. A questo scopo, assicurarsi che il *random seed* cambi ad ogni ripetizione (ad esempio, selezionare *Random seed (unique simulation runs)* in *Simulation – Randomness*)

L'analisi *what-if* dello scenario alternativo si ottiene prevedendo un percorso diverso per alcune entità passanti. Si tratta di realizzare una biforcazione attraverso il blocco *SelectOutput*, che indirizza le entità verso la prima fila di attesa con probabilità 80% (e il restante 20% direttamente verso la seconda fila d'attesa).