**Università degli Studi di Perugia**

**Dipartimento di Matematica e Informatica**



**AA 2021/2022**

**Simulazione McDrive**

**Professore**: **Studenti**:

Sergio Tasso Leonardo Angeletti

Riccardo Conti

**Indice**

1. Introduzione………………………………………………………………………3
2. Descrizione del sistema……………………………………………………..3
3. Rappresentazione del sistema……………………………………..…….4
4. Possibile soluzione………………………………………………………..….11

**1) Introduzione**

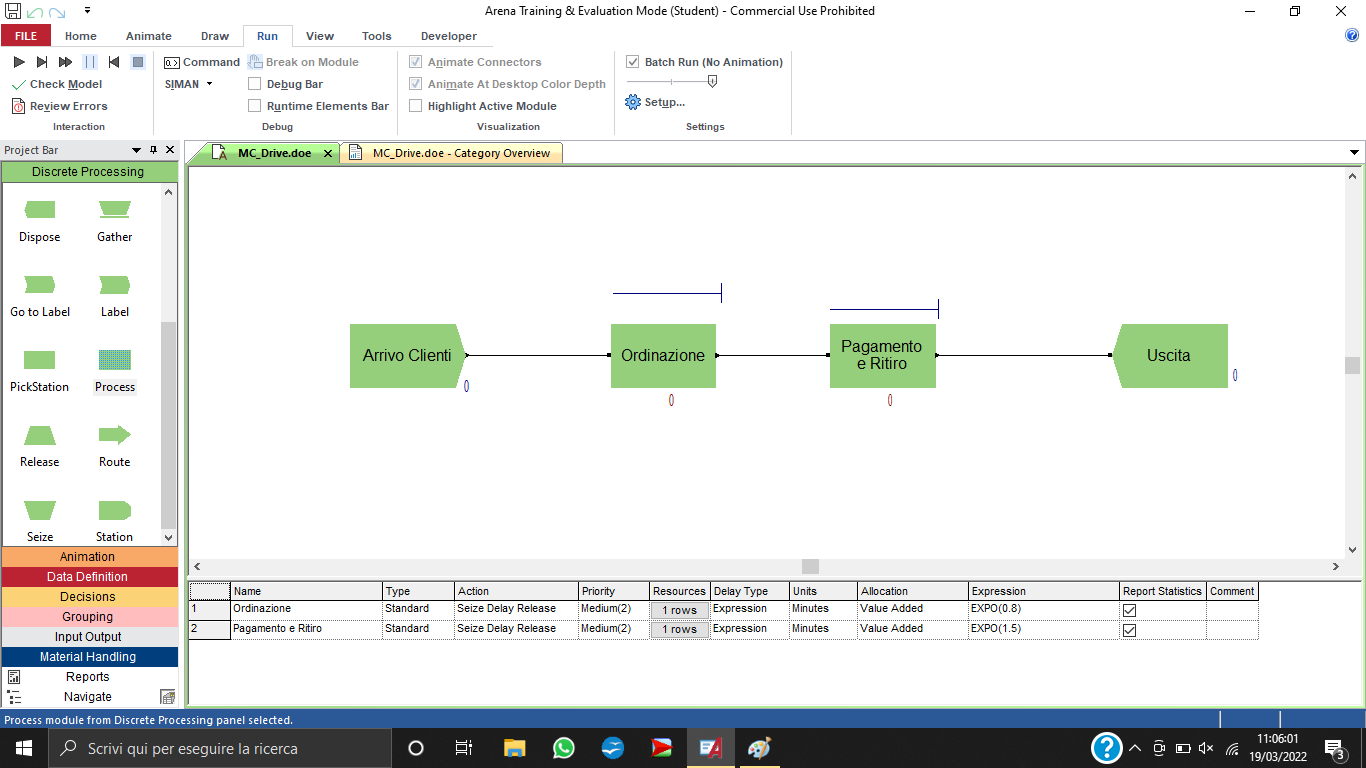
Il responsabile di un McDonald ha problemi a determinare il numero di dipendenti da attivare, considerando in particolare il servizio McDrive e l’afflusso settimanale. Avendo un nostro amico che lavora nel McDonald di Città di Castello ci siamo fatti dare il numero di scontrini effettuati dal mcDrive durante ogni giorno della settimana.

Abbiamo notato che l’affluenza dei clienti al McDrive è maggiore nei venerdì sera e nei sabato sera, e quindi siamo andati a contare il numero di auto che arrivavano durante un sabato dalle ore 18 alle 24, dividendo le auto in arrivo in intervalli di 10 min.

Abbiamo quindi stimato la disciplina delle code e i modelli di arrivo e servizio, come descritto nei paragrafi seguenti. Parte dei dati ricevuti sono stati quindi usati per sviluppare le distribuzioni empiriche, mentre i rimanenti sono stati invece utilizzati per la convalida del simulatore al 90% del livello di confidenza. Abbiamo infine riportato una proposta di soluzione al problema sopra descritto. Il modello di simulazione è stato implementato utilizzando il software Arena Simulation.

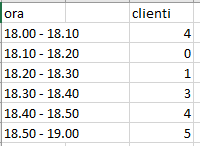
**2) Descrizione del sistema**

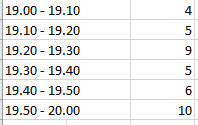
Abbiamo preso come riferimento il McDonald di città di Castello e il mcDrive funziona in questa maniera: si imbocca una strada che porta fino al punto dove si ordina nel quale gli automobilisti si fermano e un operatore segna gli ordini, per poi passare dall’altra parte dell’edificio dove si ritira ciò che si è ordinato e si paga, per poi andarsene.

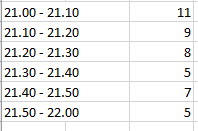
**

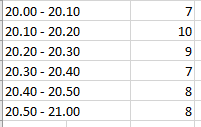
**3) Rappresentazione teorica del sistema**

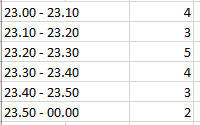
Prima di tutto vediamo come sono gli arrivi del nostro sistema, se riusciamo a trovare una qualche distribuzione matematica che seguono. Abbiamo contato gli arrivi dividendoli in intervalli di 10 minuti ciascuno.

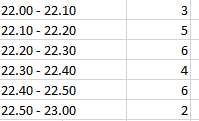






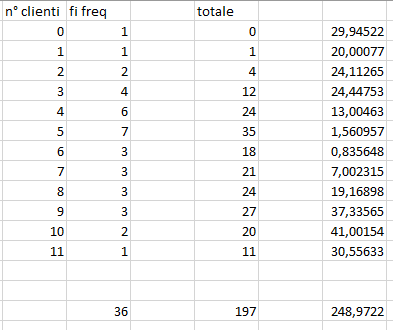






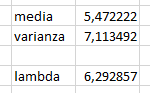


E infine abbiamo raggruppato per vedere le frequenze con le quali fossero arrivati i clienti



Calcoliamo media e varianza grazie alle somme delle ultime due colonne nello screenshot sopra per poter ipotizzare una distribuzione.

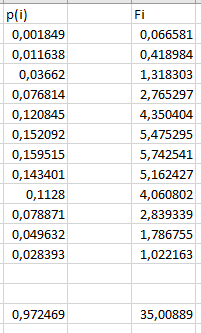
La media sarà data da 197/36, mentre la varianza sarà 248.9722/36-1 e otterremo



Dato che media e varianza non sono troppo differenti, ipotizziamo una poissoniana utilizzando come parametro λ la media tra media e varianza.

Il test che andremo a svolgere è il **Goodness Of Fit**, in quanto, avendo diviso in intervalli da 10 minuti ciascuno, abbiamo n = 36, e sappiamo che con n > 30 è più corretto utilizzare il Goodness Of Fit.

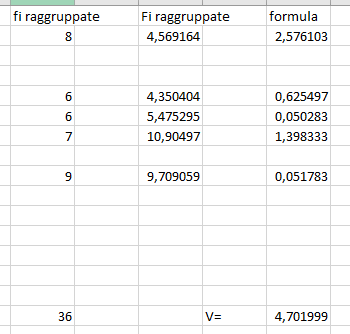
Adesso calcoliamo, per ogni fi, il valore teorico p(i) che dovremmo avere se fosse effettivamente una distribuzione poissoniana di ragione λ, dopodiché andiamo a calcolare Fi, moltiplicando ogni valore ottenuto al passo precedente per il totale delle osservazioni e quindi n.



Abbiamo quasi terminato il nostro test, ci manca solo di raggruppare le nostre frequenze creando dei campioni che contengano un fi >= 5. Decidiamo quindi di raggruppare le frequenze di arrivo di clienti 0,1,2,3, mentre 4 e 5 ne hanno 6 quindi vanno bene, uniamo 6 e 7 e infine raggruppiamo 8,9,10 e 11.

Allo stesso modo raggruppiamo anche i valori delle Fi.

Infine applichiamo la seguente formula per calcolare la nostra V:



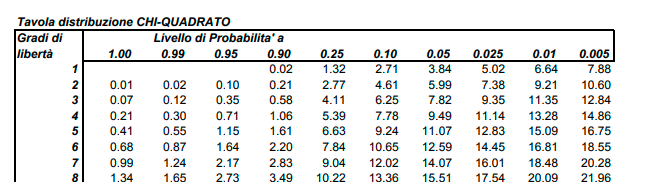
Ora sappiamo che la nostra V vale 4.70, per vedere quanti gradi di libertà abbiamo, applichiamo la seguente formula:

df = n° intervalli -1 - n° parametri della distribuzione

Nel nostro caso, dopo il raggruppamento abbiamo 5 intervalli e abbiamo 1 solo parametro (λ) e quindi

df = 5-1-1 = 3

Vediamo nella tavola del chi-quadro, con 3 gradi di libertà, dove si colloca la nostra V, ovvero 4.70.



Sappiamo quindi che la distribuzione degli arrivi segue una poissoniana perciò il nostro sistema possiede le seguenti caratteristiche:

* I tempi di arrivo seguono appunto una distribuzione poissoniana e quindi gli interarrivi sono esponenziali -> **M**
* I tempi di servizio sono distribuiti esponenzialmente -> **M**
* I dipendenti sono multipli, ma c’è solamente una postazione in cui poter ordinare quindi il sistema ha un servente singolo. -> **1**

Il sistema sarà quindi rappresentato dal modello M/M/1 con i seguenti parametri:

**Arrivi**

Durante le 6 ore in cui siamo stati lì ad osservare il sistema sono arrivate 197 persone e quindi:

* Ogni ora arrivano mediamente 33 persone
* Ogni minuto arrivano mediamente 0.55 persone

λ = 33 ore-1 o λ = min-1

**Servizio**

Mediamente il tempo per prendere le ordinazioni è stato di circa 50 secondi per cui:

* Ogni minuto viene servito 1.2 clienti
* Ogni ora vengono serviti 72 clienti

μ = 72 ore-1 o μ = min-1

**Utilizzazione**

Calcoliamo di questo sistema applicando la formula

Facciamo i conti utilizzando i valori in minuti ed otterremo il valore

Essendo minore di 1, il nostro sistema è in condizione di stazionarietà, possiamo quindi andare a calcolare alcuni valori interessanti.

Partiamo calcolando il tempo medio di attesa in coda **Tw**

Tw = (ρ/µ) / (1 − ρ)

e quindi

Vediamo ora il numero medio di utenti in coda **W**

e quindi

Una volta finita la fase di ordinazione, si avanza leggermente con l’auto fino al casello in cui si paga e si ritira il cibo, il tutto in un tempo distribuito esponenzialmente di circa 1.5 minuti. Vediamo ora come cambiano i paramentri.

= min-1  o μ = min-1

Avendo < 1 il sistema non è congestionato, calcoliamone la media di utenti in coda e il tempo medio di attesa in coda teorici.

Calcoliamo ora il tempo totale passato da un utente nel sistema, tra tempo in coda e tempo di servizio.

Ttot  = Tw1 + Ts1 + spostamento auto da sportello ordinazione a ritiro + Tw2 + Ts2

Ttot  = 0.71 *min* + 0.83 *min* + 0.42 *min* + 7.07 *min* + 1.5 *min* = 10.53 *min*

**4) Simulazione con Arena e convalida del modello**

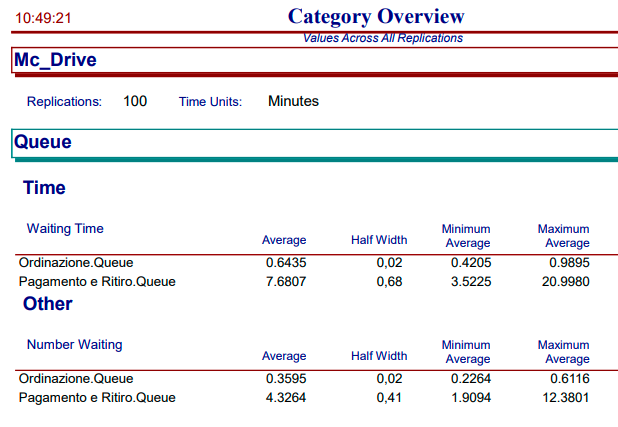
Abbiamo utilizzato il software Arena per simulare questo processo abbiamo ottenuto i seguenti valori, che non si discostano molto da ciò che avevamo previsto grazie alla teoria.

|  |  |  |
| --- | --- | --- |
| **Parametro** | **Valore Teorico** | **Valore Simulato** |
| **Tw**Ordinazioni | ≅ 0.71 *min* | ≅ 0.64 *min* |
| **Tw**Pagamento e Ritiro | ≅ 7.07 min | ≅ 7.68 *min* |
| **W** Ordinazioni | ≅ 0.56 *utenti* | ≅ 0.36 *utenti* |
| **W** Pagamento e Ritiro | ≅ 3.89 *utenti* | ≅ 4.33 *utenti* |

La colonna Half Width all’interno del report sottostante permette inoltre di determinare se il modello convalida il simulatore. Può avere tre valori:

* **Insufficient**:il numero di campioni non è sufficiente per la convalida del modello.
* **Correlated**: i campioni sono correlati tra loro mentre per convalidare il modello è necessario che i campioni siano distribuiti indipendentemente.
* **Un valore numerico**: Se invece all’interno della colonna comprare un valore numerico, significa che in almeno il 95% delle prove ripetute la media del campione viene riportata all’interno dell’intervallo ±[Half Width] della media del campione.

Nel nostro caso abbiamo Half Width con dei valori numerici per cui siamo sicuri che il nostro modello convalida al 95% di confidenza.



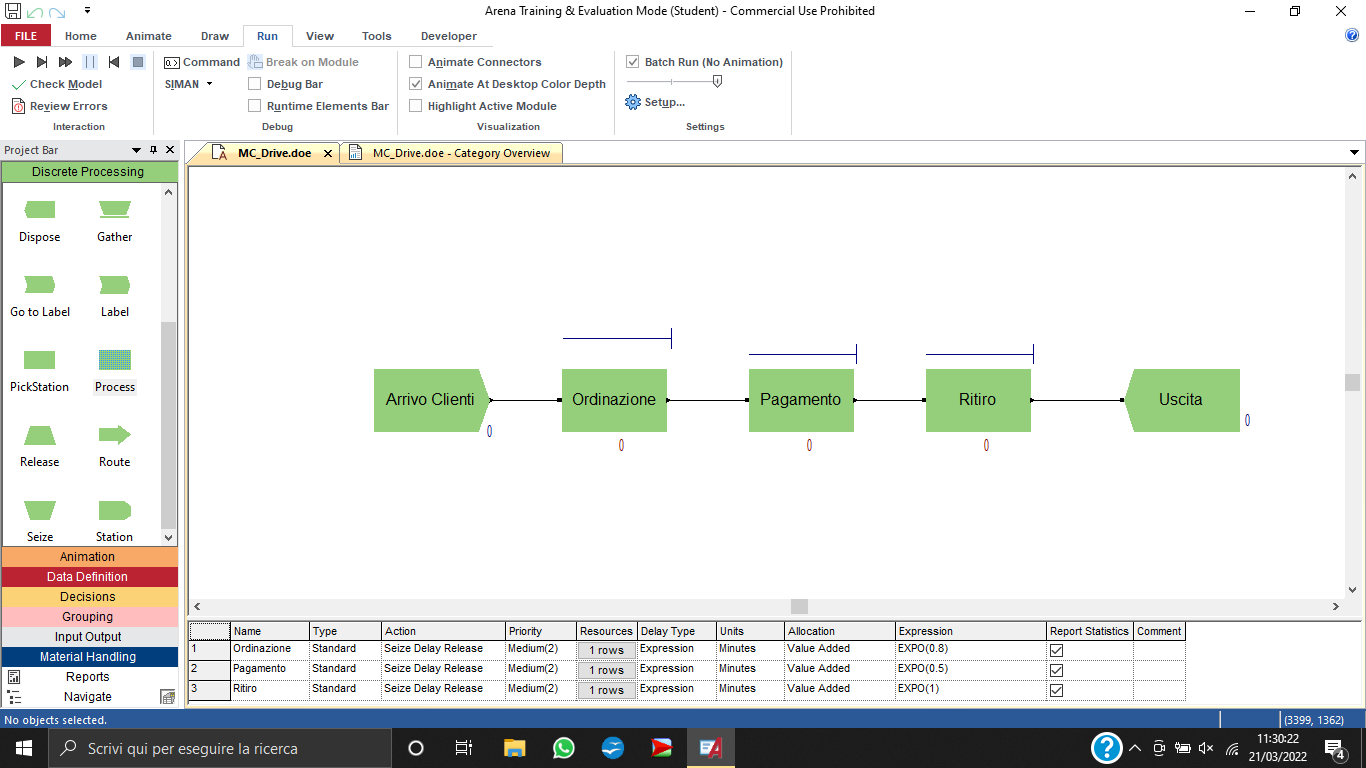
**5) Possibile Soluzione**

Abbiamo elaborato una possibile tecnica che riesca a ridurre il tempo medio totale speso nel sistema. La nostra soluzione è quella di aggiungere un casello intermedio tra quello di ordinazione e quello di ritiro del cibo in cui si può pagare dividendo i tempi di servizio in circa 30 secondi al casello per pagare e 1 minuto per ritirare il cibo. Il tempo totale speso nel sistema sarà quindi:

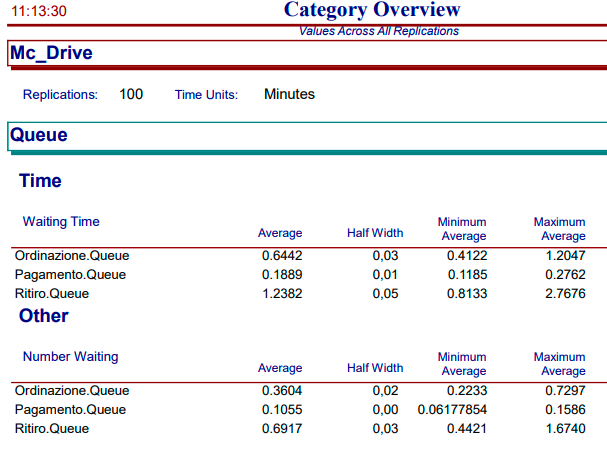
Ttot  = Tw1 + Ts1 + spostamento auto da sportello ordinazione a pagamento + Tw2 + Ts2

+ spostamento auto da sportello pagamento a ritiro + Tw3 + Ts3

Ttot  = 0.71 *min* (Tw1) + 0.83 *min* (Ts1)+ 0.42 *min* (spostamento)+ 0.19 *min* (Tw2) + 0.5 *min* (Ts2) + 0.42 *min* (spostamento )+ 1.2 *min* (Tw3)+ 1 *min* (Ts3) = 5.27 *min*



Vediamo se i tempi di attesa in coda rispettano le nostre previsioni teoriche:



|  |  |  |
| --- | --- | --- |
| **Parametro** | **Valore Teorico** | **Valore Simulato** |
| **Tw**Ordinazioni | ≅ 0.71 *min* | ≅ 0.64 *min* |
| **Tw**Pagamento | ≅ 0.19 min | ≅ 0.19 *min* |
| **Tw**Ritiro | ≅ 1.2 *min* | ≅ 1.2 *min* |
| **W** Ordinazioni | ≅ 0.56 *utenti* | ≅ 0.36 *utenti* |
| **W** Pagamento | ≅ 0.11 *utenti* | ≅ 0.11 *utenti* |
| **W** Ritiro | ≅ 0.67 *utenti* | ≅ 0.69 *utenti* |

Anche in questo caso il valore Half Width è un valore numerico per cui la simulazione è convalidata almeno al 95%, inoltre i valori emersi dalla run della simulazione sono molto vicini ai valori ipotizzati.

Ovviamente la soluzione proposta sfrutta la possibilità di avere un dipendente in più che stia nello sportello del pagamento, e non sappiamo se sia effettivamente vantaggioso economicamente per la società. Si potrebbe nel caso effettuare quest’ aggiunta nei soli momenti in cui ci sono un numero elevato di auto in coda, mentre per il tempo restante il dipendente può svolgere altre mansioni e il sistema ritorna a funzionare come ha sempre funzionato.