

Analisi delle Code agli Sportelli Comunali: Ottimizzazione dei Servizi al Cittadino



Progetto valido per il corso di Performance Modeling of Computer Systems and Networks

A.A. 22/23

Di Marco Luca – 0333083

Di Totto Luca - 0333084

Sommarario

1 - DESCRIZIONE DEL SISTEMA	3
1.1 OBIETTIVI.....	3
2 - MODELLO CONCETTUALE	5
2.1 EVENTI	7
3 - MODELLO DELLE SPECIFICHE	8
3.1 DATI IN INPUT	8
3.2 DISTRIBUZIONI	9
3.3 PROBABILITÀ DI ROUTING	10
3.3.1 <i>Centralino</i>	10
3.3.2 <i>Anagrafe + Stato Civile</i>	10
3.3.3 <i>URP + Protocollo + Cultura</i>	11
3.3.4 <i>Servizi Scolastici e Servizi Sociali</i>	11
3.3.5 <i>Matrice di Routing</i>	11
3.4 COSTI E PROFITTI	12
4 – IMPLEMENTAZIONE	12
5 – MODELLO COMPUTAZIONALE	12
5.1 STATO ED EVENTI DEL SISTEMA	12
5.1.1 <i>Clock di sistema</i>	12
5.1.2 <i>Classe Sum</i>	12
5.1.3 <i>Classe EventList</i>	13
5.1.4 <i>Classe EventHandler</i>	13
5.1.5 <i>Classi Node</i>	14
5.2 CLASSE RANDOMFUNCTION E RNGS.....	15
6 – VERIFICA	16
6.1 CENTRALINO.....	16
6.2 ANAGRAFE	17
6.3 URP	18
6.4 SERVIZI SCOLASTICI	19
6.5 SERVIZI SOCIALI	20
6.6 STATO CIVILE.....	21
6.7 PROTOCOLLO.....	22
6.7 CULTURA	23
6.8 CONTROLLI DI CONSISTENZA	24
7 – VALIDAZIONE	24
8 – ESECUZIONE DEGLI ESPERIMENTI E SIMULAZIONI	28
8.1 ANALISI DEL FLUSSO FORZATO (COLLO DI BOTTIGLIA)	28
8.2 SIMULAZIONE AD ORIZZONTE INFINITO	30
8.3 SIMULAZIONE AD ORIZZONTE FINITO.....	33
9 – VERSIONE MIGLIORATIVA	36
9.1 TEMPO DI ESECUZIONE	36
9.1.1 <i>Servizi scolastici</i>	36
9.1.2 <i>Servizi sociali</i>	37
9.1.3 <i>Cultura</i>	38
9.2 NUMERO DI SERVENTI	39

9.2.1 Servizi scolastici.....	39
9.2.2 Servizi sociali	40
9.2.3 Cultura.....	41
9.3 VERSIONE DEFINITIVA	42
9.3.1 Servizi scolastici.....	42
9.3.2 Servizi sociali	42
9.3.3 Cultura.....	43
10 – CONCLUSIONI	43

1 - Descrizione del sistema

L'obiettivo di questo progetto è la realizzazione di un modello a reti di code ispirato ad un caso di studio reale.

Il sistema reale analizzato è quello rappresentante lo scenario presente all'interno della maggior parte degli uffici comunali italiani, in particolare, viene preso come modello il comune di Lanciano, in provincia di Chieti.

Generalmente, nella maggior parte dei comuni italiani, un normale cittadino si rivolge ad uno degli sportelli dell'ufficio comunale di appartenenza per richiedere uno o più servizi offerti. Nello studio del sistema, viene ipotizzato che tutti i servizi siano fisicamente vicini l'uno con l'altro, ovvero che il cittadino, o comunque la richiesta, non impieghi del tempo nel caso in cui debba spostarsi da un ufficio ad un altro per richiedere servizi differenti.

I cittadini che si rivolgono agli sportelli possono seguire due tipi di percorsi:

- I cittadini che non conoscono dove si trova fisicamente l'ufficio richiesto, o che non sanno a quale ufficio rivolgersi per svolgere una certa mansione, si recheranno prima allo sportello della portineria (chiamato nel sistema Centralino), dove ci saranno uno o più addetti che avranno il compito di fornire indicazioni al cittadino per il problema presentato, in modo che il cittadino possa recarsi all'ufficio corretto.
- I cittadini che conoscono già il tipo di servizio da richiedere e dunque in quale edificio recarsi, evitano di passare per la portineria (o Centralino), e si dirigono direttamente all'ufficio di competenza.

Tra tutti i vari tipi di servizi che un presidio comunale può offrire, nello studio vengono presi in analisi:

- Centralino
- Anagrafe
- Stato civile
- Servizi sociali
- Servizi culturali
- Servizi scolastici
- Protocollo
- URP (Ufficio Relazioni con il Pubblico)

Si ipotizza inoltre che per ogni ufficio che offre un servizio, sia presente una sala d'attesa in cui i cittadini, nel caso in cui gli sportelli siano tutti occupati, possano mettersi in coda per essere serviti e attendere il loro turno.

Viene considerata anche l'eventualità in cui il cittadino, nel momento di arrivo nell'ufficio, possa aver dimenticato qualche documento necessario per completare la propria richiesta e decidere di abbandonare la coda e rinunciare al servizio.

Come noto, in tutti gli uffici comunali è presente il problema delle lunghe attese che scoraggiano i cittadini e creano malumori ogni qualvolta questi abbiano la necessità di rivolgersi ad uno sportello.

1.1 Obiettivi

Attualmente i tempi di attesa medi generali agli sportelli comunali risultano essere molto elevati, tanto da portare negli anni a diverse critiche e servizi d'inchiesta che denunciano quanto accade.

In particolare, nel caso di studio in esame, abbiamo rilevato, recandoci personalmente negli uffici e 'intervistando' gli addetti degli sportelli comunali presenti nello studio, che i tempi di servizio medi sono i seguenti:

Centro	$E(s)$
Centralino	1 minuto
Anagrafe	20 minuti
URP	6 minuti
Servizi scolastici	15 minuti
Servizi sociali	25 minuti
Stato civile	25 minuti
Protocollo	6.5 minuti
Cultura	120 minuti

Mentre i tempi di attesa medi riscontrati fisicamente agli sportelli sono:

Centro	$E(T_Q)$
Centralino	0.5 minuti
Anagrafe	5 minuti
URP	30 minuti
Servizi scolastici	100 minuti
Servizi sociali	30 minuti
Stato civile	10 minuti
Protocollo	1 minuto
Cultura	300 minuti

Alla luce di questo, gli obiettivi dello studio sono:

1. Creare un modello a reti di code che contenga tutti i servizi offerti dagli sportelli comunali analizzati, in modo che questo abbia valori di tempi di attesa e di servizio che si avvicinino il più possibile ai valori di attesa e servizio medio identificati per ogni singolo sportello nel comune in cui è stato effettuato lo studio.
2. Studiare il modello creato al punto 1, in modo da identificare quali possano essere le cause che, nel modello preso in esame, rallentino l'esecuzione delle procedure. Successivamente creare un modello migliorativo che cerchi di abbassare i tempi di risposta, e di conseguenza quelli di attesa media, di quei centri che riscontrano tempi eccessivamente elevati nella singola giornata lavorativa. Nel modello migliorativo, si vogliono garantire i seguenti QoS (Quality of service):

Centro	$E(T_s)$
Centralino	2 minuti
Anagrafe	30 minuti
URP	35 minuti
Servizi scolastici	60 minuti
Servizi sociali	45 minuti
Stato civile	35 minuti

Protocollo	10 minuti
Cultura	300 minuti

Nel modello migliorativo, inoltre, si vuole portare l'utilizzazione dei centri ad alto utilizzo ad un valore accettabile intorno all'80%.

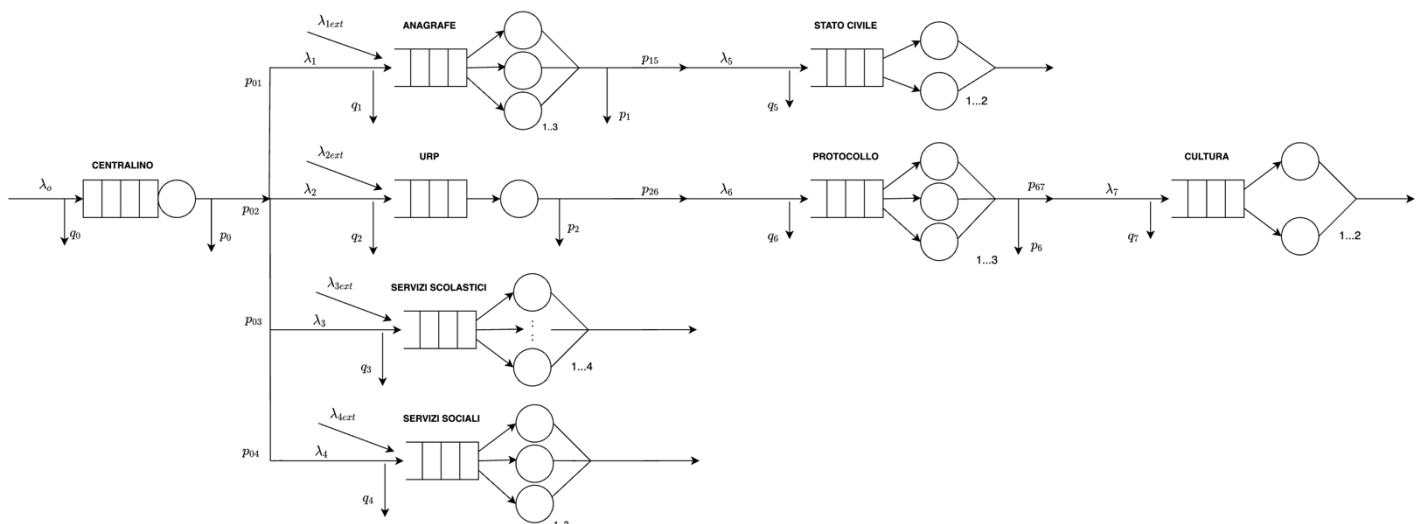
2 - Modello concettuale

Gli utenti del sistema sono i cittadini, residenti e non, che si rivolgono fisicamente agli sportelli presenti nel territorio comunale per richiedere un servizio oppure tramite richieste telefoniche o via e-mail.

Come anticipato prima, questi possono sapere già a quale ufficio rivolgersi per richiedere un certo tipo di servizio, e sono quelli che andranno direttamente in coda al centro di elaborazione interessato, oppure possono non conoscere lo sportello a cui fare riferimento, e in quel caso passeranno prima per un punto di informazioni, rappresentato dal Centralino dell'ufficio comunale.

Questi tipi di percorsi specifici rappresentano casi reali che possono verificarsi e verranno approfonditi successivamente. Ogni richiesta che un cittadino effettua ad un singolo sportello viene modellato come singolo job.

Il system diagram in esame è illustrato nella seguente figura:



Possiamo dunque notare tre differenti tipologie di flussi in ingresso:

- Nel caso del solo centralino, gli ingressi saranno solo esterni al sistema e saranno rappresentati dalla variabile: $\lambda_0 = \lambda_{totCentralino}$;
- Nel caso dei centri come Anagrafe, URP, Servizi Scolastici e Servizi Culturali troviamo due differenti flussi in ingresso: ingressi esterni ed ingressi interni (da Centralino). Per il centro i -esimo troveremo che il flusso totale in ingresso è dato da:
 - $\lambda_i + \lambda_{iext} = \lambda_{totCentroi-esimo}$;
- Nel caso dei centri come Stato Civile, Protocollo e Cultura, troviamo una sola tipologia di flusso in ingresso: una percentuale del flusso in uscita del centro precedente. La descrizione è quindi la seguente:
 - $\lambda_i = \lambda_{tot}$ dove, in questo caso, λ_i descrive solo arrivi interni.

La modellazione è la seguente:

- Centralino: M/G/1;
- Anagrafe: M/G/3;
- URP: M/G/1;
- Servizi Scolastici: M/G/4;
- Servizi Sociali: M/G/3;
- Stato Civile: M/G/2;
- Protocollo: M/G/3;
- Cultura: M/G/2.

Tutti i centri presentato degli arrivi che seguono una distribuzione esponenziale, mentre per i tempi di servizio viene utilizzata la distribuzione normale troncata, in modo da escludere i valori che si allontanano troppo dai tempi medi di servizio reali.

Ogni centro all'interno del servizio descritto, presenta una singola coda infinita, che mira a rappresentare la sala d'attesa che in genere è presente all'interno di tutti gli uffici. Ciascuna coda di ogni centro è rappresentata da uno scheduling FIFO non preemptive, come è naturale pensare che avvenga all'interno degli sportelli pubblici, dove in genere viene rispettato tra i cittadini l'ordine di arrivo in loco.

Gli N serventi presenti all'interno di ogni centro rappresentano gli operatori comunali impegnati nel fornire il servizio.

I centri che rappresentano il Centralino (o portineria) ed URP, sono rappresentati da un singolo servente in quanto, nel sistema reale, sono presidiati effettivamente da una singola persona.

Tutti gli altri centri invece sono multi-serventi poiché, nel sistema reale, per ogni servizio sono presenti più operatori impegnati ad offrire i vari servizi.

Non sono previste code distinte per nessun tipo di servizio, e di conseguenza nessun caso di priorità tra i job.

In alcuni casi è prevista la possibilità che un cittadino, dopo aver completato la richiesta presso uno sportello, si rechi direttamente in coda in un altro centro poiché può esistere una correlazione tra alcuni nodi.

- Per ciascuno degli sportelli abbiamo che ogni servente n-esimo può assumere due stati:
 - IDLE;
 - BUSY.
- Per ogni sportello i-esimo troviamo inoltre:
 - λ_i = tasso di ingresso nello sportello i-esimo da uno sportello precedente (ingresso interno);
 - λ_{iext} = tasso di ingresso nello sportello i-esimo dall'esterno (se previsto ingresso dall'esterno);
 - q_i = probabilità di uscita dallo sportello prima di essere servito.
- Gli sportelli che prevedono un possibile sportello successivo per la lavorazione della richiesta presentano inoltre:
 - p_{ij} = probabilità di andare dallo sportello i-esimo allo sportello j-esimo dopo essere stato servito dal primo;
 - p_i = probabilità di uscita dallo sportello i-esimo e conclusione della richiesta.

Nel caso di sportelli che non sono direttamente collegati ad altri, la probabilità p_i non viene indicata nel grafico, poiché si presuppone che al termine del servizio escano dal centro comunale.

Inoltre, i valori delle probabilità p_{ij} (e di conseguenza p_i) sono basati su dati reali.

In ogni centro, un job che arriva in coda, verrà subito servito se è presente almeno uno dei serventi di quel centro che si trova nello stato IDLE; altrimenti, se nessun servente è libero nell'istante di arrivo del job, questo si metterà in coda e attenderà il suo turno.

In ogni centro, inoltre, nel caso in cui in coda sia presente un numero di job diverso da 0, non può accadere che uno dei serventi sia nello stato IDLE (sistema *work-conserving*). Al contrario, nel caso in cui il numero di job nel centro sia uguale a 0, i serventi possono trovarsi nello stato IDLE o BUSY, in base al fatto che stiano servendo un job oppure no.

Nel momento in cui un centro completa la richiesta di un job, controlla la coda relativa a quel servizio; nel caso in cui la coda sia non vuota, serve il prossimo job; altrimenti, nel caso in cui la coda sia vuota, il servente entra nello stato IDLE.

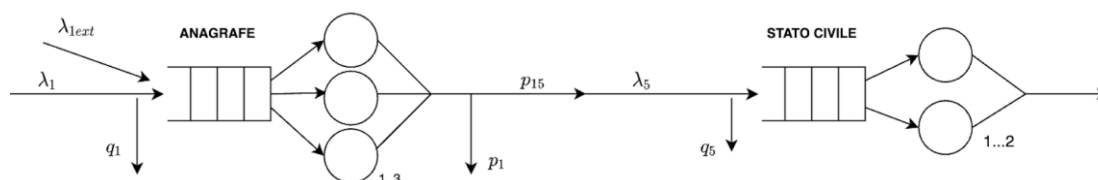
2.1 Eventi

Gli eventi possibili ad ogni istante di tempo, in tutti i centri presi in considerazione sono i seguenti:

- Completamento della richiesta del cittadino (job) presso un operatore dello sportello (servente);
- Arrivo di un nuovo job all'interno della coda se tutti i serventi sono busy, altrimenti se la coda è empty o almeno un servente è idle, il job entra direttamente in servizio;
- Abbandono di un job dal centro, per motivi come la mancanza di documenti necessari a completare la domanda;
- Se un qualsiasi servente è idle e la coda è non empty, viene servito immediatamente il prossimo job (non possiamo avere serventi idle se la coda è non vuota).

Come già anticipato, il passaggio nel Centralino non è obbligatorio: se il cittadino conosce già la locazione dello sportello può superare quest'ultimo (verrà trattato come arrivo esterno al sistema).

In specifiche coppie (o triplette) di centri, come *Anagrafe + Stato Civile*, oppure *URP + Protocollo + Cultura*, possiamo avere un evento di completamento nel primo centro e inserimento nel secondo (o terzo) centro di servizio successivo (si prende come riferimento nella foto il solo collegamento *Anagrafe + Stato Civile*):



Queste coppie o triplette di centri puntano ad astrarre ciò che avviene nella realtà: una richiesta, per essere inoltrata al centro Stato Civile deve essere necessariamente elaborata dal centro Anagrafe; così come per le richieste per i servizi di Cultura, devono necessariamente essere prima protocollate dal centro Protocollo, per poi essere inoltrate al centro di interesse.

Si definisce, infine, lo stato del sistema come l'insieme dei singoli stati di ogni centro. In quest'ultimo, qualsiasi esso sia, verranno memorizzati:

- il numero di job presenti nel sistema in un certo istante;

- il numero di job in ingresso dall'interno;
- il numero di job in ingresso dall'esterno (se previsti);
- il numero di job in coda;
- il numero di job serviti (specificato per ogni singolo servente);
- il numero di job che hanno abbandonato il centro;
- lo stato di ogni specifico servente (idle o busy);
- l'istante di tempo corrente e l'istante di tempo in cui si verifica il prossimo evento

Come già anticipato, il completamento di un job all'interno di un servente può portare o alla sua uscita dal sistema o all'inserimento verso un secondo centro (a seconda della fisionomia del nodo specifico).

La gestione degli eventi e della simulazione segue una logica next-event: tutti gli eventi da processare sono inseriti all'interno di una struttura dati ed eseguiti secondo il loro istante di tempo. Nel caso di centri che sono solo successivi ad altri (si veda il caso Stato Civile rispetto Anagrafe), verranno simulati solo successivamente al completamento della simulazione in Anagrafe, così da poter popolare il centro con i job che non sono usciti dal sistema, ma inviati al centro successivo; anche se simulato successivamente al centro precedente, gli istanti di tempo vengono rispettati e simulati coerentemente con lo scorrere del tempo.

3 - Modello delle Specifiche

Per rendere il più possibile realistica la modellazione del sistema in studio, siamo riusciti ad ottenere direttamente dagli uffici del comune di Lanciano, come anticipato nel punto 1.1 *Obiettivi*, i dati riguardanti i tempi di servizio medi per singolo job (o richiesta) dei vari centri.

3.1 Dati in input

Ricordiamo che i dati forniti non sono estremamente precisi, ma seguono più una linea generale e sono basati su dei valori medi, questo perché anche gli uffici erano sprovvisti di dati estremamente accurati.

Centro	Tempi di servizio
Anagrafe	20 minuti
Stato civile	25 minuti
Centralino	1 minuti
Servizi sociali	25 minuti
Servizi culturali	120 minuti
Servizi scolastici	15 minuti
Protocollo	5 minuti
Ufficio Relazioni con il Pubblico	5 minuti

Inoltre, abbiamo potuto ottenere i dati riguardanti il numero di arrivi di richieste (o job) ad ogni centro ipotizzando l'orario di lavoro pari a 390 minuti (ossia 6 ore e 30) giornalieri:

Centro	Numero di job/giorno
Anagrafe	50
Stato civile	30
Centralino	200
Servizi sociali	50
Servizi culturali	25

Servizi scolastici	150
Protocollo	50
Ufficio Relazioni con il Pubblico	70

Da questi dati abbiamo ricavato i seguenti tassi in ingresso dall'esterno ai centri che prevedono un arrivo dall'esterno:

- Centralino: $\lambda_0 = 0.51 \text{ j/min}$
- Anagrafe: $\lambda_1 = 0.05 \text{ j/min}$
- URP: $\lambda_2 = 0.085 \text{ j/min}$
- Servizi Scolastici: $\lambda_3 = 0.205 \text{ j/min}$
- Servizi Sociali: $\lambda_4 = 0.065 \text{ j/min}$

Dai dati sopra citati, abbiamo potuto ricavare i seguenti tempi di servizio medi per ogni servente di ogni centro:

Centro	Media (secondi)	Deviazione	Troncamento sx	Troncamento dx
Centralino	1	0.40	0.10	2
Anagrafe	20	5	5	35
Servizi Scolastici	15	4	10	25
Servizi Sociali	25	10	10	40
Stato Civile	25	5	10	35
Protocollo	6.5	2	3	10
Cultura	120	30	60	240

Ad eccezione del centro URP che presenta tempi di servizio divisi per probabilità:

Probabilità	Media (secondi)	Deviazione	Troncamento sx	Troncamento dx
0.30	5	2	2	8
0.30	3	1.5	1	5
0.06	15	5	8	20
0.04	17.5	5	10	25
0.30	7.5	2.5	3	12

Questo poiché l'URP riceve tipi di richieste molto differenti tra loro che richiedono, quindi, tempi di servizio diversi. Le percentuali sono state recuperati dai dati reali sul campo.

3.2 Distribuzioni

Dopo aver studiato in maniera approfondita il caso di studio presentato, abbiamo tratto le seguenti conclusioni:

- Per simulare correttamente quello che è lo scenario reale, si è scelto di utilizzare la distribuzione normale troncata per la generazione dei tempi di servizio poiché permette di eliminare valori fuori dal normale ed improbabili al fine di riflettere al meglio il funzionamento del sistema reale. Inoltre, permette di rimuovere tempi di servizi negativi, in quanto non rappresentativi della realtà.

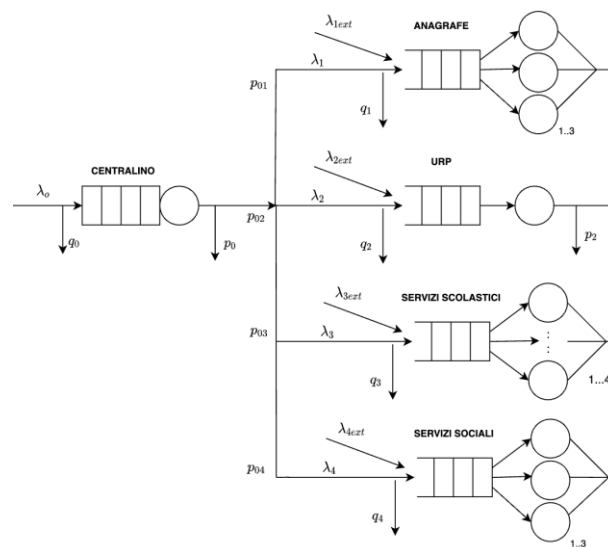
- La distribuzione presa in considerazione per i tempi di interarrivo è la distribuzione esponenziale. Questo perché essa permette di modellare correttamente gli arrivi casuali ed indipendenti con frequenza costante dei cittadini presso uno sportello. Si presuppone, infatti, basandosi sui dati reali, che nelle varie fasce orarie non ci siano particolari momenti di picco o momenti di arrivo quasi nulli o nulli.

3.3 Probabilità di Routing

Per la definizione delle probabilità di routing si usano i dati statistici recuperati all'interno del comune preso in considerazione.

3.3.1 Centralino

In uscita dal centralino troviamo cinque possibili diramazioni: Anagrafe, URP, Servizi Scolastici, Servizi Culturali o l'uscita dal sistema, come descritto dalla seguente figura:



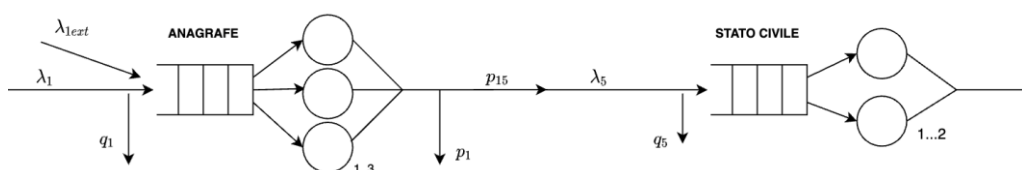
Dai dati raccolti sul campo troviamo i seguenti valori giornalieri:

- Probabilità del job di muoversi verso l'ufficio Anagrafe: $p_{01} = 0.1$;
- Probabilità del job di muoversi verso l'ufficio URP: $p_{02} = 0.15$;
- Probabilità del job di muoversi verso l'ufficio Servizi Scolastici: $p_{03} = 0.4$;
- Probabilità del job di muoversi verso l'ufficio Servizi Sociali: $p_{04} = 0.125$.

Infine, la probabilità di abbandono dal Centralino (e conseguente uscita dal sistema) viene descritta dalla variabile $q_0 = 0.02$.

3.3.2 Anagrafe + Stato Civile

Prendendo in considerazione la sola catena di centri Anagrafe + Stato Civile, come mostrato nella figura successiva, troviamo i seguenti valori (che rispecchiano i dati empirici raccolti sul campo):



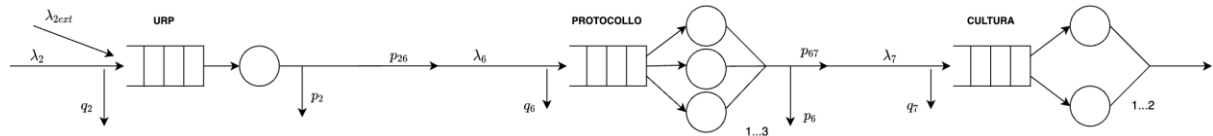
- Probabilità del job di muoversi da Anagrafe verso Stato civile: $p_{15} = 0.625$;
- Di conseguenza, la probabilità di uscire dal sistema dopo essere stato servito da Anagrafe senza entrare in Stato Civile: $p_1 = 1 - p_{15} = 0.375$.

Infine:

- La probabilità di abbandono dal centro Anagrafe viene descritta dalla variabile: $q_1 = 0.05$;
- La probabilità di abbandono dal centro Stato Civile viene descritta dalla variabile: $q_5 = 0.02$.

3.3.3 URP + Protocollo + Cultura

Prendendo in considerazione la sola catena di centri URP + Protocollo + Cultura, come mostrato nella figura successiva, troviamo i seguenti valori (che rispecchiano i dati empirici raccolti sul campo):



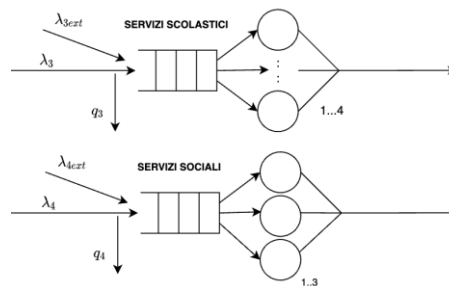
- La probabilità del job di muoversi da URP verso Protocollo: $p_{26} = 0.7575$;
- Di conseguenza, la probabilità di uscire dal sistema dopo essere stato servito da URP senza entrare in Protocollo: $p_2 = 1 - p_{26} = 0.2425$.
- La probabilità del job di muoversi da Protocollo verso Cultura: $p_{67} = 0.35$;
- Di conseguenza, la probabilità di uscire dal sistema dopo essere stato servito da Protocollo senza entrare in Cultura: $p_6 = 1 - p_{67} = 0.65$.

Infine:

- La probabilità di abbandono dal centro URP viene descritta dalla variabile: $q_2 = 0.01$;
- La probabilità di abbandono dal centro Protocollo viene descritta dalla variabile: $q_6 = 0.01$;
- La probabilità di abbandono dal centro Cultura viene descritta dalla variabile: $q_7 = 0.02$.

3.3.4 Servizi Scolastici e Servizi Sociali

Prendendo in considerazione i soli due centri Servizi Scolastici e Servizi Sociali, come mostrato nella figura successiva, non ci sono probabilità di routing verso altri centri: tutti i job entranti in un servente vengono serviti ed escono dal sistema.



Prendiamo, quindi, solo in considerazione le probabilità di abbandono dal sistema:

- La probabilità di abbandono dal centro Servizi Scolastici viene descritta dalla variabile: $q_3 = 0.03$;
- La probabilità di abbandono dal centro Servizi Culturali viene descritta dalla variabile: $q_4 = 0.04$.

3.3.5 Matrice di Routing

Per riassumere quanto detto precedentemente, la matrice di routing è la seguente:

	0	1	2	3	4	5	6	7
0	0	0.1	0.15	0.4	0.125	0	0	0
1	0	0	0	0	0	0.625	0	0
2	0	0	0	0	0	0	0.757	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0.35
7	0	0	0	0	0	0	0	0

Sulle righe sono rappresentati i centri da cui il job termina il servizio ed esce dallo stesso e sulle colonne sono rappresentati i centri verso i quali il job può successivamente entrare per essere servito.

3.4 Costi e profitti

Nel caso preso in considerazione, essendo un servizio statale, non vengo presi in considerazione possibili costi di mantenimento.

4 – Implementazione

Il modello sopra descritto è stato implementato utilizzando come linguaggio di programmazione Java: la scelta deriva dal fatto che un ambiente come un sistema a code in cui i vari centri interagiscono tra loro, può essere meglio rappresentato da una programmazione ad oggetti.

5 – Modello Computazionale

Per la simulazione del sistema si è adottato l'approccio Next-Event Simulation che permette al sistema di lavorare correttamente e simulare il sistema reale.

5.1 Stato ed eventi del sistema

5.1.1 Clock di sistema

Per tenere traccia dell'evoluzione del tempo si utilizza una classe *Time* per ogni istanza di *Node*. La classe *Time* mantiene informazioni riguardanti il tempo corrente ed il tempo dell'evento successivo. Fornisce inoltre tutti i metodi utili per modificare e recuperare i valori di questi ultimi.

```
public class Time {
    private double current;
    private double next;
```

- Il valore di *current* indica il tempo corrente di simulazione;
- Il valore *next* indica il tempo del prossimo evento (il più imminente).

5.1.2 Classe Sum

La classe *Sum* viene utilizzata per mantenere informazioni su quanti job sono stati serviti da un servente e per quanto tempo ha effettuato servizio. Viene implementato un array di istanze di questa classe per ogni centro, tramite i quali è possibile tenere traccia delle informazioni sopra citate per ogni singolo servente del nodo. L'array contiene tante istanze di *Sum* quanti sono i serventi del centro. Fornisce inoltre tutti i metodi utili per modificare e recuperare i valori sopra indicati.

```
public class Sum {
    double service; /*service times*/
    long served; /*number served*/
```

- Il valore di *service* indica il tempo di servizio del servente;

- Il valore di *served* indica il numero di job serviti dal servernte.

5.1.3 Classe EventList

La classe *EventList* mantiene le informazioni degli eventi del centro preso in considerazione. Ogni Nodo ha un suo array di *EventList* con grandezza variabili:

- Per i nodi che prevedono l'ingresso di job esclusivamente dall'esterno (nel nostro caso, il Centralino), l'array avrà dimensione pari al numero di servernti + 2 di cui:
 - La entry 0-esima verrà utilizzata per tenere traccia del prossimo arrivo dall'esterno;
 - Le entry da 1 a #server saranno utilizzate per tenere traccia dello stato di ogni servernte i-esimo (0 = idle, 1 = busy);
 - L'ultima entry verrà utilizzata per tenere traccia dell'evento di abbandono dal centro.
- Per i nodi che prevedono l'ingresso di job sia dall'interno che dall'esterno, l'array avrà dimensione pari al numero di servernti + 3 di cui:
 - La entry 0-esima verrà utilizzata per tenere traccia del prossimo arrivo dall'esterno;
 - Le entry da 1 a #server saranno utilizzate per tenere traccia dello stato di ogni servernte i-esimo (0 = idle, 1 = busy);
 - La penultima entry verrà utilizzata per tenere traccia dell'evento di abbandono dal centro.
 - L'ultima entry verrà utilizzata per tenere traccia dell'evento di arrivo di un job dall'interno (da un altro centro).
- Per i nodi che prevedono solo ingressi di job dall'interno, viene utilizzata la stessa struttura precedentemente descritta, dove però la entry 0-esima rimane inutilizzata.

```
public class EventList {
    private double t;
    private int x;
```

- Il valore di *t* indica il tempo dell'evento;
- Il valore di *x* indica lo stato dell'evento o del servernte: nel caso delle entry diverse da quelle del servernte, questa indica se c'è l'imminente arrivo di un evento (o viceversa); nel caso delle entry corrispondenti ai servernti, indica lo stato idle o busy.

All'interno della stessa classe troviamo, oltre ai metodi per recuperare e modificare i valori sopra citati, anche il metodo *NextEvent()* che permette il recupero dell'evento più imminente, mediante la ricerca all'interno dell'array *EventList* dello specifico centro chiamante.

```
public static int NextEvent(EventList[] event, int servers) {
    int e;
    int i = 0;

    while (event[i].x == 0)
        i++;
    e = i;
    while (i < (servers + 1)) {
        i++;
        if ((event[i].x == 1) && (event[i].t < event[e].t))
            e = i;
    }
    return (e);
}
```

5.1.4 Classe EventHandler

La classe *EventHandler* mantiene lo stato interno del sistema: viene istanziata come classe *Singleton*, per mantenere correttezza nella gestione degli eventi. Al suo interno:

- Mantiene una configurazione del numero dei servernti per ogni centro;

- Istanza, mantiene ed aggiorna il valore degli array di *EventList* per tutti i centri;
- Istanza, mantiene ed aggiorna il valore degli *ArrayList* di eventi dall'interno per tutti i centri;
- Istanza e mantiene il valore della matrice di routing precedentemente descritta;
- Mantiene i valori delle probabilità di abbandono dai centri.

Fornisce, quindi, metodi per la gestione dei punti sopra citati. Un estratto è il seguente:

```
//ANAGRAFE
private EventList[] eventNodo1;
private ArrayList<Double> internalArrivalNodo1;
private static int server1 = 3;
```

5.1.5 Classi Node

Tutti i nodi del sistema sono un'istanza di quattro possibili tipologie di classi Java chiamate *Node*, dove ognuna descrive una determinata tipologia di sportello comunale:

- *Node1*: classe utilizzata per la descrizione degli sportelli che prendono job in ingresso solo dall'esterno e possono inviare i job sia all'esterno del sistema che verso i successivi nodi (come il Centralino);
- *Node2*: classe utilizzata per la descrizione degli sportelli che prendono job in ingresso sia dagli sportelli precedenti che dall'esterno e permettono ai job, dopo essere stati serviti, soltanto di uscire dal sistema (come per il Servizio Scolastico o Servizio Sociale);
- *Node3*: classe utilizzata per la descrizione degli sportelli che prendono in ingresso job solo da sportelli precedenti e permettono agli stessi sia di uscire dal sistema, sia di entrare in un successivo sportello, dopo essere stati serviti (come per il Protocollo);
- *Node4*: classe utilizzata per la descrizione degli sportelli che prendono job in ingresso da sportelli precedenti e dall'esterno e permettono sia di uscire dal sistema, sia di entrare in uno sportello successivo, dopo essere stati serviti (come per l'Anagrafe).

All'interno della classe vengono mantenuti i dati utili all'esecuzione del centro come:

- *START*: indica l'istante di inizio dell'esecuzione;
- *STOP*: indica l'istante di fine dell'esecuzione;
- *area*: indica il tempo totale speso dai job all'interno del centro. Viene utilizzato per il calcolo delle statistiche;
- *external_num_job*: indica il numero di job in ingresso al centro dall'esterno;
- *num_job*: numero di job presenti in un certo istante di tempo nel centro;
- *internal_num_job*: indica il numero di job in ingresso al centro dall'interno;
- *server*: indica il numero di serventi del centro;
- *jobServiti*: indica il numero di job serviti nella simulazione dal centro;
- *name*: indica il nome del centro;
- Istanze di classi *Time* e *Sum* ed *EventHandler*, per la gestione dei dati e degli eventi nel centro;
- *exitProbability*: indica la probabilità di uscita dalla coda; il suo valore viene recuperato all'interno di *EventHandler*;
- *id*: indica l'id numero del nodo utile per la sua identificazione.

Ogni classe viene istanziata e successivamente eseguita mediante l'invocazione del metodo *batchMeans()*; quest'ultimo prevede la possibilità di eseguire la simulazione in due modi differenti, dipendentemente dal valore di due variabili booleane *batch* e *finiteHorizon*. Nel caso in cui *batch* sia settata true, viene eseguita la simulazione ad orizzonte infinito mediante la tecnica di batch means (si

veda successivamente); altrimenti, in caso *batch* sia false e *finiteHorizon* true, verrà eseguita la simulazione standard rappresentativa del caso reale (orizzonte finito)

I metodi che permettono l'esecuzione della simulazione prendono il nome di *normalWork()* oppure di *finiteHorizon()*. La seconda esegue la simulazione fintanto che ci sono arrivi (esterni o interni) con un tempo inferiore al valore di *STOP*, oppure fintanto che sono presenti job all'interno del sistema. Di seguito un estratto del metodo *normalWork()*:

```
while ((batch) ? (job_batch < 2048) : ((this.handler.getEventNode(id)[0].getX() != 0) || (this.num_job > 0))) {

    EventList[] eventList = this.handler.getEventNode(id);

    e = EventList.NextEvent(eventList, server);
    this.time.setNext(eventList[e].getT());
    this.area = this.area + (this.time.getNext() - this.time.getCurrent()) * this.num_job;
    this.time.setCurrent(this.time.getNext());

    if(!batch) {
        if (eventList[0].getT() > this.STOP) { //se il
            eventList[0].setX(0);
            this.handler.setEventNode(id, eventList);
        }
    }
}
```

Il metodo *normalWork()* quindi, scandisce l'array *EventList* per trovare l'evento prossimo più imminente da eseguire e lo esegue aggiornando le variabili di stato e di tempo.

All'interno di quest'ultimo vengono usati metodi come *getService()* e *getArrival()* per la generazione di arrivi o servizi, forniti dalla classe *RandomFunction* che verrà descritta successivamente.

Rimandiamo la lettura all'apposito capitolo 8.2 per l'esecuzione in modalità Batch Means.

Il comportamento di *finiteHorizon()* è simile a quello di *normalWork()* (rimandiamo al capitolo 8.3 "Simulazione ad orizzonte finito" per maggiori dettagli).

5.2 Classe RandomFunction e Rngs

La classe *RandomFunction* viene utilizzata per configurare i tempi di interarrivo e di servizio dei vari centri. Anche questa classe viene istanziata come *Singleton*.

Al suo interno troviamo metodi per la generazione di tempi di servizio e di arrivo seguendo le distribuzioni opportune utilizzando la classe *Rngs.java* per la generazione dei numeri casuali: quest'ultima fornisce dei metodi per la generazione di numeri casuali mediante la metodologia del generatore causale di Lehmer.

A questo proposito viene utilizzato uno specifico seed per tutte le simulazioni dello studio ed è pari a 12345.

Ad ogni centro vengono riservati due diversi stream: uno per la generazione degli arrivi ed uno per la generazione dei servizi ad ogni centro, per un totale di 16 stream. Un singolo stream viene riservato per la generazione delle probabilità.

La classe fornisce un metodo *getService()* per la generazione dei tempi di servizio ed un metodo *getJobArrival()* per la generazione dei tempi di arrivo. In entrambi, la generazione differisce a seconda del centro che invoca il metodo (differenziato a seconda dell'*id*).

Un estratto del metodo *getService()* è il seguente:


```
//ANAGRAFE
case 1:
    try {
        rngs.selectStream(id+8);
        departure = NormalTruncated(20, 5, 5, 35);
        //departure = NormalTruncated(10, 2.5, 5, 80);
        return departure;
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Un estratto del metodo *getJobArrival()* è il seguente:

```
public double getJobArrival(int id) {
    //arrival += this.start;
    rngs.selectStream(id);
    //arrival += Exponential(intTime);
    arrival[id] += Exponential(interarrival[id]);
    return arrival[id];
}
```

6 – Verifica

Nella fase di verifica, l'obiettivo è controllare che i risultati delle simulazioni siano coerenti e confrontarli con i valori teorici. Quest'ultimi sono stati ricavati simulando il comportamento del sistema con tempi di servizio esponenziali, invece che normali: i valori di quest'ultimi sono stati configurati in modo da avvicinare il più possibile il comportamento del sistema a quello reale, per quanto possibile.

Si desidera, quindi, verificare che l'implementazione del modello computazionale sia effettivamente corretta.

Viene effettuato un confronto per ogni centro tra i valori del modello analitico e quelli ottenuti dalla simulazione ad orizzonte infinito mediante il metodo di batch means. Quest'ultimo viene impostato con i seguenti parametri:

- Centralino b=2048 e k=50;
- Tutti gli altri centri b=200 e k=50;

Rimandiamo al capitolo Batch Means per i dettagli implementativi.

6.1 Centralino

Il tasso di arrivo al sistema è calcolato come il reciproco del tempo di interarrivo, ed è quindi:

$$\lambda_s = \frac{1}{1.96} = 0.51 \text{ job/minuto}$$

A questo, considerando la probabilità di uscita dal centro, si ottiene il tasso di ingresso finale:

$$\lambda_0 = 0.4998 \text{ job/minuto}$$

Tramite il tempo medio di servizio del servente $E(S) = 1 \text{ minuto}$ ricaviamo l'utilizzazione:

$$\rho = \frac{\lambda}{\mu} = \lambda * E(S)$$

Riportiamo nella seguente tabella i risultati ottenuti seguendo il modello analitico:

Risultati del nodo <i>Centralino (M/G/1)</i> attraverso il modello analitico	
Utilizzazione (ρ)	0.499
Numero medio di job in coda $E(N_q)$	0.497
Numero medio di job nel centro $E(N_s)$	0.997
Tempo di attesa medio in coda $E(T_q)$	0.996
Tempo di attesa medio nel sistema $E(T_s)$	1.996

Risultati del nodo <i>Centralino (M/G/1)</i> prodotto dalla simulazione	
Utilizzazione (ρ)	0.50 ± 0.00
Numero medio di job in coda $E(N_q)$	0.50 ± 0.02
Numero medio di job nel centro $E(N_s)$	1.01 ± 0.02
Tempo di attesa medio in coda $E(T_q)$	1.00 ± 0.04
Tempo di attesa medio nel sistema $E(T_s)$	2.00 ± 0.05

6.2 Anagrafe

Il tasso di arrivo al centro è calcolato come il reciproco del tempo di interarrivo per i job esterni più il tasso di uscita del centro precedente, ed è quindi:

$$\lambda_{1ext} = \frac{1}{20} = 0.05$$

$$\lambda_s = \lambda_{1ext} + (\lambda_0 * p_{01}) = 0.05 + (0.4998 * 0.1) = 0.0999 \text{ job/minuto}$$

A questo, considerando la probabilità di uscita dal centro, si ottiene il tasso di ingresso finale:

$$\lambda_1 = 0.0949 \text{ job/minuto}$$

Tramite il tempo medio di servizio del servente i -esimo $E(S_i) = 20 \text{ minuti}$ ricaviamo l'utilizzazione:

$$\rho = \frac{\lambda}{m\mu} = \lambda * \frac{E(S_i)}{m}$$

Riportiamo nella seguente tabella i risultati ottenuti seguendo il modello analitico:

Risultati del nodo <i>Anagrafe (M/G/3)</i> attraverso il modello analitico	
Utilizzazione (ρ)	0.632
Numero medio di job in coda $E(N_q)$	0.682
Numero medio di job nel centro $E(N_s)$	2.583
Tempo di attesa medio in coda $E(T_q)$	7.187
Tempo di attesa medio nel sistema $E(T_s)$	27.187

Risultati del nodo <i>Anagrafe (M/G/3)</i> prodotto dalla simulazione	
Utilizzazione (ρ)	0.65 ± 0.02
Numero medio di job in coda $E(N_q)$	0.63 ± 0.08
Numero medio di job nel centro $E(N_s)$	2.58 ± 0.12
Tempo di attesa medio in coda $E(T_q)$	6.49 ± 0.77
Tempo di attesa medio nel sistema $E(T_s)$	26.55 ± 0.99

6.3 URP

Il tasso di arrivo al centro è calcolato come il reciproco del tempo di interarrivo per i job esterni più il tasso di uscita del centro precedente, ed è quindi:

$$\lambda_{2ext} = \frac{1}{11.76} = 0.085$$

$$\lambda_s = \lambda_{2ext} + (\lambda_1 * p_{02}) = 0.085 + (0.4998 * 0.15) = 0.16 \text{ job/minuto}$$

A questo, considerando la probabilità di uscita dal centro, si ottiene il tasso di ingresso finale:

$$\lambda_2 = 0.1584 \text{ job/minuto}$$

Tramite il tempo medio di servizio del servente $E(S) = 5 \text{ minuti}$ ricaviamo l'utilizzazione:

$$\rho = \frac{\lambda}{\mu} = \lambda * E(S)$$

Riportiamo nella seguente tabella i risultati ottenuti seguendo il modello analitico:

Risultati del nodo <i>URP (M/G/1)</i> attraverso il modello analitico	
Utilizzazione (ρ)	0.792
Numero medio di job in coda $E(N_q)$	3.016
Numero medio di job nel centro $E(N_s)$	3.807
Tempo di attesa medio in coda $E(T_q)$	19.038
Tempo di attesa medio nel sistema $E(T_s)$	24.038

Risultati del nodo <i>URP (M/G/1)</i> prodotto dalla simulazione	
Utilizzazione (ρ)	0.79 ± 0.02
Numero medio di job in coda $E(N_q)$	2.86 ± 0.79
Numero medio di job nel centro $E(N_s)$	3.65 ± 0.81
Tempo di attesa medio in coda $E(T_q)$	17.78 ± 4.58
Tempo di attesa medio nel sistema $E(T_s)$	22.81 ± 4.64

6.4 Servizi Scolastici

Il tasso di arrivo al centro è calcolato come il reciproco del tempo di interarrivo per i job esterni più il tasso di uscita del centro precedente, ed è quindi:

$$\lambda_{3ext} = \frac{1}{4.878} = 0.205$$

$$\lambda_s = \lambda_{3ext} + (\lambda_2 * p_{03}) = 0.205 + (0.4998 * 0.4) = 0.405 \text{ job/minuto}$$

A questo, considerando la probabilità di uscita dal centro, si ottiene il tasso di ingresso finale:

$$\lambda_3 = 0.393 \text{ job/minuto}$$

Tramite il tempo medio di servizio del servente $E(S_i) = 5 \text{ minuti}$ ricaviamo l'utilizzazione:

$$\rho = \frac{\lambda}{m\mu} = \lambda * \frac{E(S_i)}{m}$$

Riportiamo nella seguente tabella i risultati ottenuti seguendo il modello analitico:

Risultati del nodo Servizi Scolastici (M/G/4) attraverso il modello analitico	
Utilizzazione (ρ)	0.491
Numero medio di job in coda $E(N_q)$	0.159
Numero medio di job nel centro $E(N_s)$	2.122
Tempo di attesa medio in coda $E(T_q)$	0.406
Tempo di attesa medio nel sistema $E(T_s)$	5.406

Risultati del nodo Servizi Scolastici (M/G/4) prodotto dalla simulazione	
Utilizzazione (ρ)	0.49 ± 0.02
Numero medio di job in coda $E(N_q)$	0.13 ± 0.03
Numero medio di job nel centro $E(N_s)$	2.13 ± 0.06
Tempo di attesa medio in coda $E(T_q)$	0.33 ± 0.07
Tempo di attesa medio nel sistema $E(T_s)$	5.32 ± 0.14

6.5 Servizi Sociali

Il tasso di arrivo al centro è calcolato come il reciproco del tempo di interarrivo per i job esterni più il tasso di uscita del centro precedente, ed è quindi:

$$\lambda_{4ext} = \frac{1}{15.385} = 0.06499$$

$$\lambda_s = \lambda_{4ext} + (\lambda_3 * p_{04}) = 0.06499 + (0.4998 * 0.125) = 0.1274 \text{ job/minuto}$$

A questo, considerando la probabilità di uscita dal centro, si ottiene il tasso di ingresso finale:

$$\lambda_4 = 0.1223 \text{ job/minuto}$$

Tramite il tempo medio di servizio del servente $E(S_i) = 20 \text{ minuti}$ ricaviamo l'utilizzazione:

$$\rho = \frac{\lambda}{m\mu} = \lambda * \frac{E(S_i)}{m}$$

Riportiamo nella seguente tabella i risultati ottenuti seguendo il modello analitico:

Risultati del nodo <i>Servizi Sociali (M/G/3)</i> attraverso il modello analitico	
Utilizzazione (ρ)	0.815
Numero medio di job in coda $E(N_q)$	2.966
Numero medio di job nel centro $E(N_s)$	5.413
Tempo di attesa medio in coda $E(T_q)$	24.261
Tempo di attesa medio nel sistema $E(T_s)$	44.265

Risultati del nodo <i>Servizi Sociali (M/G/3)</i> prodotto dalla simulazione	
Utilizzazione (ρ)	0.83 ± 0.02
Numero medio di job in coda $E(N_q)$	3.54 ± 0.96
Numero medio di job nel centro $E(N_s)$	6.04 ± 1.02
Tempo di attesa medio in coda $E(T_q)$	27.81 ± 7.22
Tempo di attesa medio nel sistema $E(T_s)$	47.88 ± 7.56

6.6 Stato Civile

Il tasso di arrivo al centro è calcolato questa volta solo come il tasso di arrivo interno, ovvero quello proveniente dal centro precedente, poiché l'ingresso a questo centro prevede obbligatoriamente il passaggio per quello precedente:

$$\lambda_{5ext} = 0$$

$$\lambda_s = \lambda_{5ext} + (\lambda_1 * p_{15}) = 0 + (0.0949 * 0.625) = 0.05931 \text{ job/minuto}$$

A questo, considerando la probabilità di uscita dal centro, si ottiene il tasso di ingresso finale:

$$\lambda_5 = 0.05813 \text{ job/minuto}$$

Tramite il tempo medio di servizio del servente $E(S_i) = 25 \text{ minuti}$ ricaviamo l'utilizzazione:

$$\rho = \frac{\lambda}{m\mu} = \lambda * \frac{E(S_i)}{m}$$

Riportiamo nella seguente tabella i risultati ottenuti seguendo il modello analitico:

Risultati del nodo <i>Stato Civile (M/G/2)</i> attraverso il modello analitico	
Utilizzazione (ρ)	0.726
Numero medio di job in coda $E(N_q)$	1.624
Numero medio di job nel centro $E(N_s)$	3.077
Tempo di attesa medio in coda $E(T_q)$	27.948
Tempo di attesa medio nel sistema $E(T_s)$	52.948

Risultati del nodo <i>Stato Civile (M/G/2)</i> prodotto dalla simulazione	
Utilizzazione (ρ)	0.76 ± 0.03
Numero medio di job in coda $E(N_q)$	1.90 ± 0.41
Numero medio di job nel centro $E(N_s)$	3.43 ± 0.45
Tempo di attesa medio in coda $E(T_q)$	30.68 ± 5.91
Tempo di attesa medio nel sistema $E(T_s)$	55.79 ± 6.12

6.7 Protocollo

Il tasso di arrivo al centro è calcolato come il tasso di arrivo interno, ovvero quello proveniente dal centro precedente, poiché l'ingresso a questo centro prevede obbligatoriamente il passaggio per quello precedente:

$$\lambda_{6ext} = 0$$

$$\lambda_s = \lambda_{6ext} + (\lambda_2 * p_{26}) = 0 + (0.1584 * 0.7575) = 0.12 \text{ job/minuto}$$

A questo, considerando la probabilità di uscita dal centro, si ottiene il tasso di ingresso finale:

$$\lambda_6 = 0.108 \text{ job/minuto}$$

Tramite il tempo medio di servizio del servente $E(S_i) = 6.5 \text{ minuti}$ ricaviamo l'utilizzazione:

$$\rho = \frac{\lambda}{m\mu} = \lambda * \frac{E(S_i)}{m}$$

Riportiamo nella seguente tabella i risultati ottenuti seguendo il modello analitico:

Risultati del nodo <i>Protocollo (M/G/3)</i> attraverso il modello analitico	
Utilizzazione (ρ)	0.234
Numero medio di job in coda $E(N_q)$	0.011
Numero medio di job nel centro $E(N_s)$	0.713
Tempo di attesa medio in coda $E(T_q)$	0.105
Tempo di attesa medio nel sistema $E(T_s)$	6.605

Risultati del nodo <i>Protocollo (M/G/3)</i> prodotto dalla simulazione	
Utilizzazione (ρ)	0.26 ± 0.01
Numero medio di job in coda $E(N_q)$	0.02 ± 0.01
Numero medio di job nel centro $E(N_s)$	0.79 ± 0.03
Tempo di attesa medio in coda $E(T_q)$	0.13 ± 0.05
Tempo di attesa medio nel sistema $E(T_s)$	6.66 ± 0.18

6.7 Cultura

Il tasso di arrivo al centro è calcolato come il tasso di arrivo interno, ovvero quello proveniente dal centro precedente, poiché l'ingresso a questo centro prevede obbligatoriamente il passaggio per quello precedente:

$$\lambda_{7ext} = 0$$

$$\lambda_s = \lambda_{7ext} + (\lambda_6 * p_{67}) = 0 + (0.108 * 0.35) = 0.0378 \text{ job/minuto}$$

A questo, considerando la probabilità di uscita dal centro, si ottiene il tasso di ingresso finale:

$$\lambda_6 = 0.03704 \text{ job/minuto}$$

Tramite il tempo medio di servizio del servente $E(S_i) = 20 \text{ minuti}$ ricaviamo l'utilizzazione:

$$\rho = \frac{\lambda}{m\mu} = \lambda * \frac{E(S_i)}{m}$$

Riportiamo nella seguente tabella i risultati ottenuti seguendo il modello analitico:

Risultati del nodo <i>Cultura (M/G/2)</i> attraverso il modello analitico	
Utilizzazione (ρ)	0.371
Numero medio di job in coda $E(N_q)$	0.117
Numero medio di job nel centro $E(N_s)$	0.868
Tempo di attesa medio in coda $E(T_q)$	3.174
Tempo di attesa medio nel sistema $E(T_s)$	23.176

Risultati del nodo <i>Cultura (M/G/2)</i> prodotto dalla simulazione	
Utilizzazione (ρ)	0.43 ± 0.03
Numero medio di job in coda $E(N_q)$	0.16 ± 0.08
Numero medio di job nel centro $E(N_s)$	1.01 ± 0.13
Tempo di attesa medio in coda $E(T_q)$	3.57 ± 1.46
Tempo di attesa medio nel sistema $E(T_s)$	23.66 ± 1.91

6.8 Controlli di consistenza

Per ogni centro del sistema, sono stati effettuati i seguenti controlli di consistenza:

1. Il primo controllo eseguito concerne i tempi di risposta nei centri:

$$E(T_s) = E(T_q) + E(S)$$

2. Il secondo controllo riguarda invece la popolazione nei centri, verificando per ogni centro la seguente equazione:

$$\begin{aligned} E(N_q) &= \lambda * E(T_q) \\ E(N_s) &= \lambda * E(T_s) \end{aligned}$$

7 – Validazione

La validazione è stata condotta mediante l'analisi del comportamento del sistema al variare di alcuni parametri, tale da verificare la corretta reazione dello stesso. I parametri che sono stati variati sono i seguenti:

- Aumentare e diminuire il tempo di servizio $E(S)$ dei serventi per verificare se si ottiene un aumento o una diminuzione dei parametri osservati;
- Aumentare o diminuire il tasso di arrivo al Centralino λ_0 per verificare se si ottiene un aumento o una diminuzione dei parametri osservati.

Di seguito la variazione del comportamento quando ad ogni centro, il tasso di servizio medio viene raddoppiato o dimezzato, rispetto al valore di partenza:

Centralino	$E(S) = 1$	$E(S) = 0.5$	$E(S) = 2$
ρ	0.51 ± 0.00	0.26 ± 0.00	0.95 ± 0.01
$E(T_s)$	1.56 ± 0.02	0.60 ± 0.00	16.80 ± 1.17
$E(N_s)$	0.79 ± 0.02	0.31 ± 0.00	7.89 ± 0.57

Anagrafe	$E(S_i) = 20$	$E(S_i) = 10$	$E(S_i) = 40$
ρ	0.62 ± 0.01	0.33 ± 0.01	0.90 ± 0.01
$E(T_s)$	23.55 ± 0.40	10.22 ± 0.06	97.56 ± 4.47
$E(N_s)$	2.22 ± 0.07	1.02 ± 0.02	6.73 ± 0.33

URP	$E(S) = 6.075$	$E(S) = 3.037$	$E(S) = 12.150$
ρ	0.89 ± 0.01	0.49 ± 0.01	0.99 ± 0.00
$E(T_s)$	32.71 ± 2.13	5.11 ± 0.16	210.40 ± 7.65
$E(N_s)$	4.76 ± 0.32	0.83 ± 0.03	16.30 ± 0.52

Scolastico	$E(S_i) = 15$	$E(S_i) = 7.5$	$E(S_i) = 30$
ρ	0.98 ± 0.00	0.77 ± 0.01	0.98 ± 0.00
$E(T_s)$	121.31 ± 3.16	10.20 ± 0.17	436.17 ± 6.06
$E(N_s)$	30.08 ± 0.78	4.03 ± 0.09	54.47 ± 0.76

Servizi sociali	$E(S_i) = 25$	$E(S_i) = 12.5$	$E(S_i) = 50$
ρ	0.88 ± 0.01	0.52 ± 0.01	0.95 ± 0.00
$E(T_s)$	52.68 ± 2.51	13.70 ± 0.16	245.35 ± 7.51
$E(N_s)$	5.65 ± 0.30	1.72 ± 0.04	14.06 ± 0.43

Stato civile	$E(S_i) = 25$	$E(S_i) = 12.5$	$E(S_i) = 30$
ρ	0.66 ± 0.01	0.36 ± 0.01	0.89 ± 0.01
$E(T_s)$	32.73 ± 0.92	12.80 ± 0.08	146.15 ± 6.16
$E(N_s)$	1.78 ± 0.08	0.74 ± 0.02	5.32 ± 0.24

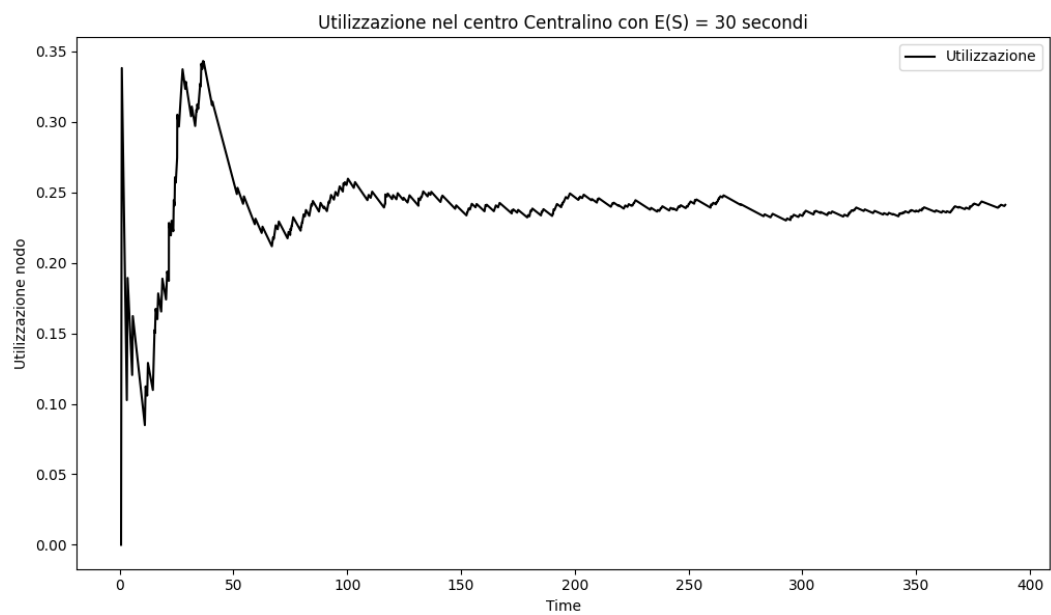
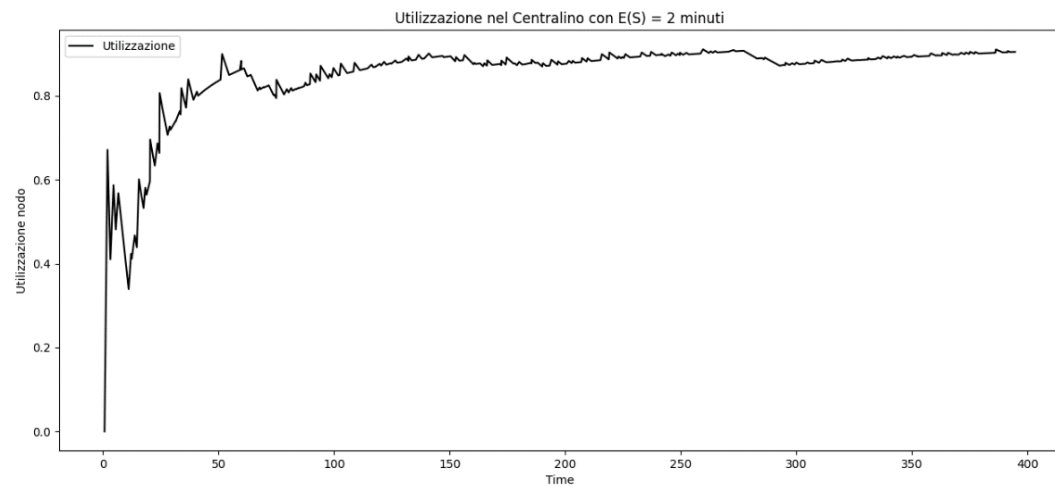
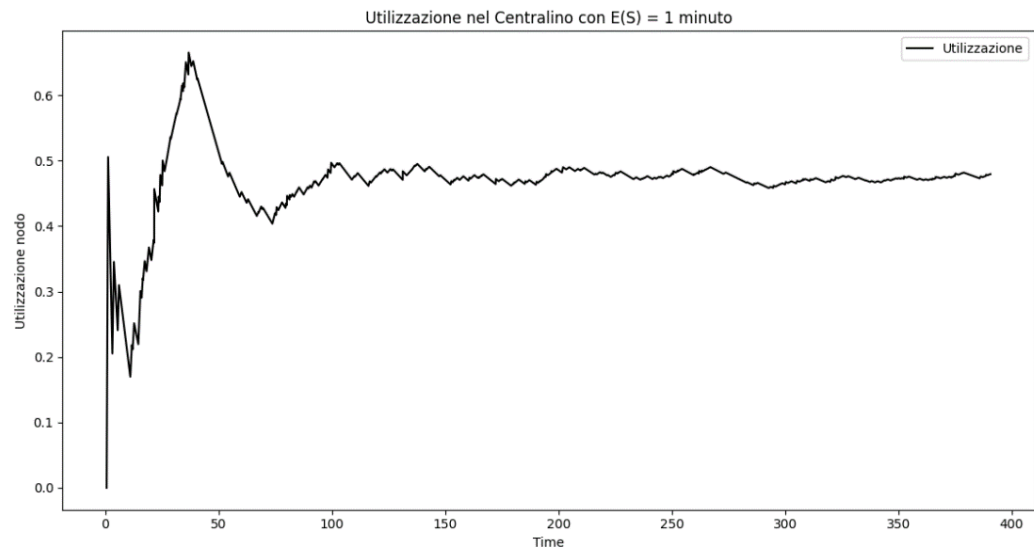
Protocollo	$E(S_i) = 6.5$	$E(S_i) = 3.25$	$E(S_i) = 13$
ρ	0.23 ± 0.00	0.12 ± 0.00	0.46 ± 0.01
$E(T_s)$	6.52 ± 0.03	3.26 ± 0.01	13.30 ± 0.07
$E(N_s)$	0.70 ± 0.01	0.35 ± 0.01	1.42 ± 0.02

Cultura	$E(S_i) = 120$	$E(S_i) = 60$	$E(S_i) = 240$
ρ	0.91 ± 0.01	0.82 ± 0.01	0.94 ± 0.0
$E(T_s)$	389.90 ± 14.59	120.41 ± 5.47	960.17 ± 29.28
$E(N_s)$	5.89 ± 0.23	3.33 ± 0.18	7.47 ± 0.23

Di seguito invece il variare del comportamento dei centri quando il tasso di ingresso al centro *Centralino* raddoppia o diminuisce rispetto al valore di partenza:

	ρ			$E(T_s)$			$E(N_s)$		
	$\lambda_0 = 0.51$	$\lambda_0 = 0.25$	$\lambda_0 = 1.02$	$\lambda_0 = 0.51$	$\lambda_0 = 0.25$	$\lambda_0 = 1.02$	$\lambda_0 = 0.51$	$\lambda_0 = 0.25$	$\lambda_0 = 1.02$
Centralino	0.51	0.26	0.96	1.56	1.19	10.72	0.79	0.31	10.28
Anagrafe	0.62	0.48	0.84	23.55	21.45	34.40	2.22	1.55	4.40
URP	0.89	0.73	0.97	32.71	16.81	91.91	4.76	2.03	14.34
Scolastico	0.98	0.96	0.99	121.31	52.41	263.67	30.08	12.75	66.07
Servizi Sociali	0.88	0.73	0.96	52.68	34.20	120.49	5.65	3.07	13.86
Stato civile	0.66	0.52	0.83	32.73	28.93	48.68	1.78	1.25	3.33
Protocollo	0.23	0.19	0.26	6.52	6.51	6.51	0.70	0.57	0.77
Cultura	0.91	0.86	0.94	389.90	280.49	528.04	5.89	4.05	8.16

Di seguito, vengono riportati alcuni dei grafici rappresentativi di alcuni dei parametri visti sopra al variare delle condizioni del sistema:



8 – Esecuzione degli esperimenti e simulazioni

La fase di esecuzione delle simulazioni si compone di tre passaggi:

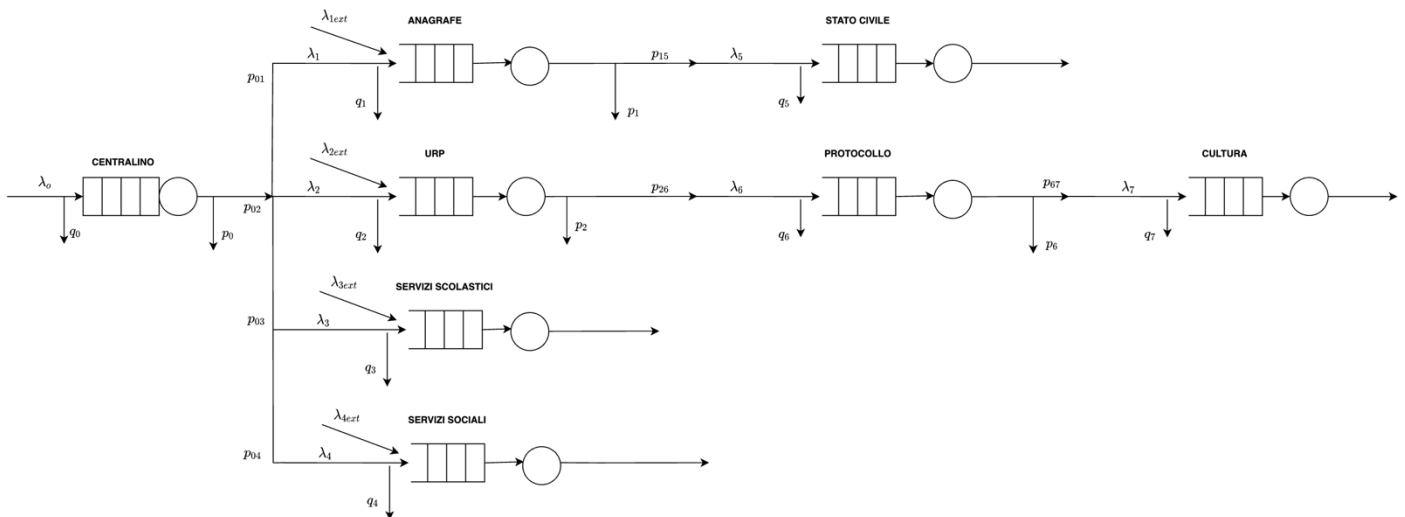
- Analisi del flusso forzato (collo di bottiglia);
- Simulazione ad orizzonte infinito;
- Simulazione ad orizzonte finito.

L'analisi del flusso forzato permette di calcolare, mediante metodi analitici, le visite medie e la domanda media per ogni centro al fine di determinare i possibili colli di bottiglia del sistema.

Le simulazioni che seguono sono state effettuate sia ad orizzonte finito che orizzonte infinito: nel nostro caso di studio, però, la simulazione ad orizzonte infinito non è identificativo del sistema poiché nei nostri obiettivi vogliamo minimizzare e studiare i tempi del sistema nell'arco ristretto alla giornata lavorativa.

8.1 Analisi del flusso forzato (Collo di bottiglia)

Viene riportato nuovamente lo schema del sistema inserito precedentemente, considerando ogni centro come singolo server:



Con $\gamma = 0.51 \text{ j/min}$ si intende il flusso di ingresso nel sistema dal centro Centralino.

Il tempo di servizio medio per ogni centro è, dunque:

- Centralino: $s_0 = 1 \text{ min}$
- Anagrafe: $s_1 = 20 \text{ min}$
- URP: $s_2 = 6.075 \text{ min}$
- Servizi Scolastici: $s_3 = 15 \text{ min}$
- Servizi Sociali: $s_4 = 25 \text{ min}$
- Stato Civile: $s_5 = 25 \text{ min}$
- Protocollo: $s_6 = 6.5 \text{ min}$
- Cultura: $s_7 = 120 \text{ min}$

Per il calcolo delle visite ad ogni centro è necessario valutare il tasso di ingresso in ognuno tenendo conto degli arrivi esterni e della probabilità di uscita:

- Centralino: $\lambda_0 = \gamma - (\gamma * q_0) = 0.500 \text{ j/min}$

Per i successivi centri vengono calcolati dei mintermini rappresentativi del valore in ingresso al centro da un centro precedente, considerando la probabilità di routing, al fine di agevolare la lettura dei calcoli successivi:

- $\delta_1 = \lambda_0 * p_{01} = 0.05 \text{ j/min}$
- $\delta_2 = \lambda_0 * p_{02} = 0.075 \text{ j/min}$
- $\delta_3 = \lambda_0 * p_{03} = 0.200 \text{ j/min}$
- $\delta_4 = \lambda_0 * p_{04} = 0.062 \text{ j/min}$

È possibile calcolare gli ingressi ai centri 1,2,3,4 mediante l'utilizzo di questi mintermini:

- Anagrafe: $\lambda_1 = (\delta_1 + \lambda_{1ext}) - ((\delta_1 + \lambda_{1ext}) * q_1) = 0.095 \text{ j/min}$
- URP: $\lambda_2 = (\delta_2 + \lambda_{2ext}) - ((\delta_2 + \lambda_{2ext}) * q_2) = 0.158 \text{ j/min}$
- Servizi Scolastici: $\lambda_3 = (\delta_3 + \lambda_{3ext}) - ((\delta_3 + \lambda_{3ext}) * q_3) = 0.397 \text{ j/min}$
- Servizi Sociali: $\lambda_4 = (\delta_4 + \lambda_{4ext}) - ((\delta_4 + \lambda_{4ext}) * q_4) = 0.127 \text{ j/min}$

Per il calcolo dei successivi tassi di ingresso bisogna calcolare nuovamente i mintermini precedenti dei centri rimanenti:

- $\delta_5 = \lambda_1 * p_{15} = 0.059 \text{ j/min}$
- $\delta_6 = \lambda_2 * p_{26} = 0.119 \text{ j/min}$

Da cui possiamo trovare:

- Stato Civile: $\lambda_5 = \delta_5 - (\delta_5 * q_5) = 0.058 \text{ j/min}$
- Protocollo: $\lambda_6 = \delta_6 - (\delta_6 * q_6) = 0.118 \text{ j/min}$

Infine, per l'ultimo centro Cultura troviamo:

- $\delta_7 = \lambda_6 * p_{67} = 0.041 \text{ j/min}$
- Cultura: $\lambda_7 = \delta_7 - (\delta_7 * q_7) = 0.040 \text{ j/min}$

Da questi valori possiamo calcolare le visite nei centri:

- Centralino: $v_0 = \frac{\lambda_0}{\gamma} = 0.98$
- Anagrafe: $v_1 = \frac{\lambda_1}{\gamma + \lambda_{1ext}} = 0.170$
- URP: $v_2 = \frac{\lambda_2}{\gamma + \lambda_{2ext}} = 0.265$
- Servizi Scolastici: $v_3 = \frac{\lambda_3}{\gamma + \lambda_{3ext}} = 0.555$
- Servizi Sociali: $v_4 = \frac{\lambda_4}{\gamma + \lambda_{4ext}} = 0.220$
- Stato Civile: $v_5 = \frac{\lambda_5}{\gamma + \lambda_{1ext}} = 0.103$
- Protocollo: $v_6 = \frac{\lambda_6}{\gamma + \lambda_{2ext}} = 0.198$
- Cultura: $v_7 = \frac{\lambda_7}{\gamma + \lambda_{2ext}} = 0.067$

Mediante i valori delle visite è possibile calcolare il valore della domanda per ogni centro, così da individuare i colli di bottiglia:

- Centralino: $D_0 = v_0 * s_0 = 0.980 \text{ min}$
- Anagrafe: $D_1 = v_1 * s_1 = 3.400 \text{ min}$
- URP: $D_2 = v_2 * s_2 = 1.609 \text{ min}$

- Servizi Scolastici: $D_3 = v_3 * s_3 = 8.325 \text{ min}$
- Servizi Sociali: $D_4 = v_4 * s_4 = 5.500 \text{ min}$
- Stato Civile: $D_5 = v_5 * s_5 = 2.575 \text{ min}$
- Protocollo: $D_6 = v_6 * s_6 = 1.287 \text{ min}$
- Cultura: $D_7 = v_7 * s_7 = 8.040 \text{ min}$

Dai risultati ottenuti è possibile notare come il collo di bottiglia del sistema è rappresentato dai centri Cultura, Servizi Sociali e Servizi Scolastici. Questi centri saranno oggetti di studio nella parte migliorativa.

8.2 Simulazione ad orizzonte infinito

L'obiettivo della simulazione ad orizzonte infinito è ottenere statistiche del sistema a stato stazionario. I prossimi dati, dunque, mostrano le statistiche ottenute simulando il sistema per un tempo molto lungo e superiore alla giornata lavorativa di 390 minuti.

Durante la simulazione si è tenuto costante il flusso in ingresso ai centri, poiché è lecito pensare che in un tempo molto lungo gli arrivi si mantengano costanti.

La simulazione viene eseguita mediante la tecnica dei Batch Means: la simulazione completa è stata divisa in k batch, ognuno di dimensione b job. A seguito di ogni esecuzione vengono azzerate le statistiche di output, mentre lo stato del sistema viene mantenuto. Per ogni esecuzione vengono presi i valori in output e calcolate le statistiche mostrate di seguito.

La tecnica del Batch Means viene utilizzata in modo da eliminare l'influenza che lo stato iniziale del sistema ha sulle statistiche di output finali; questo metodo, infatti, si basa sul calcolare le statistiche medie per ognuno dei k batch, in modo da poter effettuare alla fine una stima complessiva, calcolata sulla base dei valori ottenuti in tutti i batch.

Per determinare le dimensioni ottimali dei batch (quindi, i valori di b e k) sono state seguite le linee guida di Banks, Carson, Nelson e Nicol:

- *"Banks, Carson, Nelson, and Nicol (2001, page 438) recommend that the batch size be increased until the lag one autocorrelation between batch means is less than 0.2."*

In particolare, mediante la raccolta dei dati in output e la classe `Acs.Java` abbiamo modificato i valori di b e k affinché l'autocorrelazione del campione (per ogni statistica) fosse inferiore a 0.2 per il $\text{lag } j = 1$. Con questo metodo sono stati trovati i seguenti valori ottimali:

- Centralino: $b = 8192, k = 50$;
- Tutti gli altri centri: $b = 200; k = 50$.

La scelta dei valori per il Centralino si diversifica dalla scelta degli altri centri poiché si è reso necessario simulare il Centralino con un numero di job maggiore rispetto agli altri centri, in modo da poter popolare in maniera sufficiente gli `ArrayList` degli arrivi interni dei centri successivi ed ottenere delle statistiche consone ed accettabili poiché basate su un numero accettabile di valori.

La motivazione è la seguente: le statistiche di output di ogni simulazione vengono raccolte ed inviate alla classe `Estimate.Java` per calcolare l'intervallo di stima al 95%; a questo proposito, se gli `ArrayList` non venissero popolati con un numero di job accettabili, si otterrebbero dei risultati non consistenti e con intervalli di stima non veritieri.

In qualsiasi caso, i job generati in uscita dal Centralino e successivi al completamento di tutti i batch di un qualsiasi centro successivo, non vengono elaborati e considerati nella stima.

La simulazione ad orizzonte infinito viene considerata per completezza nello studio del sistema, ma non riflette gli obiettivi poiché non è una simulazione rappresentativa del sistema reale. Questa si limita a mostrare i tempi di risposta e l'utilizzazione del sistema a stato stazionario.

Simulando il sistema è possibile però, notare il non raggiungimento della stazionarietà: nello specifico, analizzando i dati del centro Cultura possiamo trovare valori fuori norma, quali:

- $E(T_s) = 51755.22 \text{ min};$
- $E(N_s) = 833.87;$
- $E(N_q) = 833.88;$
- $E(T_q) = 51633.23 \text{ min};$
- $\rho = 1.$

I valori appena mostrati sono evidentemente fuori equilibrio ed il sistema, per il centro considerato, non raggiunge valori di stazionarietà.

Il non raggiungimento della stazionarietà influisce negativamente anche sul calcolo delle autocorrelazioni, che per alcuni valori non è possibile portare al di sotto del valore di 0.2 anche con diverse coppie di valori b e k.

Un altro centro che non raggiunge la stazionarietà è Servizi Sociali. Nello specifico, i suoi valori medi sono:

- $E(T_s) = 626.43 \text{ min};$
- $E(N_s) = 75.11;$
- $E(N_q) = 75.12;$
- $E(T_q) = 601.37 \text{ min};$
- $\rho = 1.$

Anche in questo caso, il non raggiungimento della stazionarietà influisce negativamente sull'autocorrelazioni dei valori raccolti.

Un ultimo centro che non raggiunge la stazionarietà è rappresentato da Servizi Scolastici e presenta i seguenti valori medi:

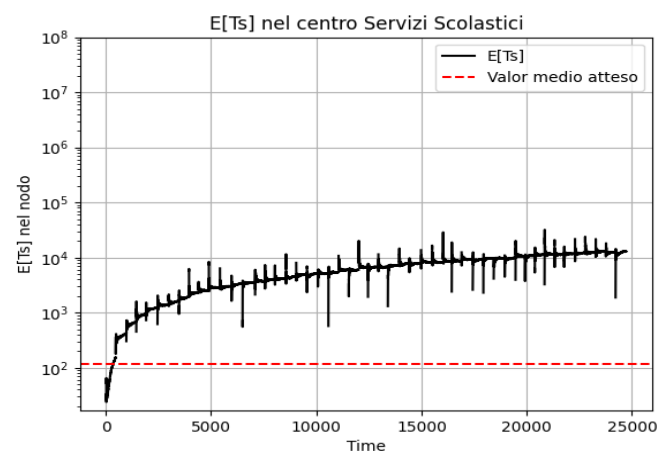
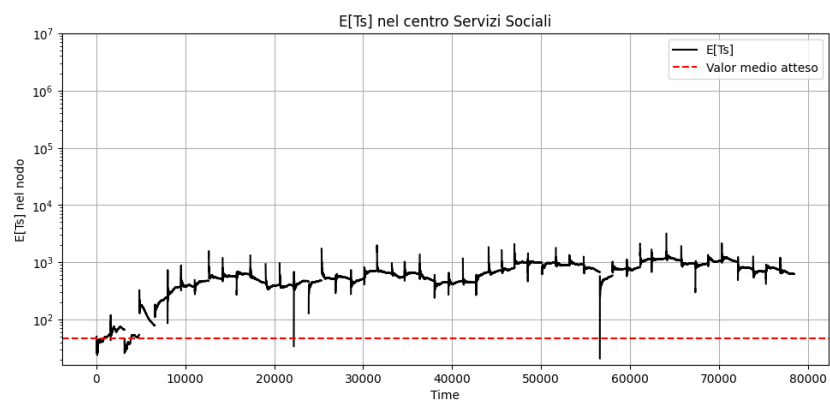
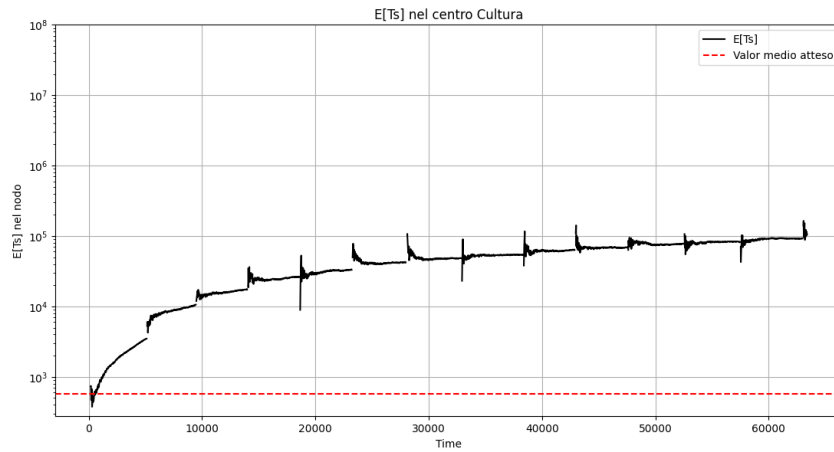
- $E(T_s) = 6602.47 \text{ min};$
- $E(N_s) = 1675.80;$
- $E(N_q) = 1671.80;$
- $E(T_q) = 6586.70 \text{ min};$
- $\rho = 1.$

Anche in questo caso, il non raggiungimento della stazionarietà influisce negativamente sull'autocorrelazioni dei valori raccolti.

I restanti centri non presentano valori fuori norma e hanno un $\rho < 1$, con valori di autocorrelazione minore di 0.2 per $\text{lag } j = 1$.

I risultati appena mostrati permettono di verificare il risultato del capitolo dell'Analisi del Flusso Forzato (si veda capitolo 8.1), in cui si mostra che i centri rappresentanti un collo di bottiglia al sistema sono, di fatto, Cultura, Servizi Sociali e Servizi Scolastici.

Si mostrano di seguito alcuni grafici dimostrativi dei valori sopra indicati:



Come è possibile notare dalle figure, i valori ottenuti da una simulazione ad orizzonte infinito sono ben oltre il valor medio atteso e non riescono a stabilizzarsi intorno ad esso. Il sistema, dunque, non raggiunge la stazionarietà.

8.3 Simulazione ad orizzonte finito

La simulazione ad orizzonte finito è stata effettuata per una durata di 390 minuti (6 ore e 30 minuti) per rappresentare la durata di una giornata lavorativa comunale. Con l'obiettivo di condurre un'analisi statistica dei risultati ottenuti, si è scelto di simulare ad orizzonte finito mediante la tecnica delle Replicazioni: l'esecuzione della singola simulazione (del lavoro dei singoli centri) è stata replicata 256 volte generando un insieme di 256 punti. Quest'ultimi sono stati utilizzati dalla classe *Estimate.Java* per generare una stima delle medie con intervallo di confidenza pari al 95% e permettere di definire un range di valori accettabili. Ogni singola run di simulazione termina nell'istante di *STOP* pari a 390 minuti (6 ore e 30 minuti – una giornata lavorativa). Successivamente, i 256 punti sono stati utilizzati per la generazione dei grafici inseriti in basso.

Come da linee guida, la simulazione con la tecnica delle Replicazioni richiede di inizializzare il seed per la generazione dei valori random all'esterno del ciclo di simulazione, per evitare che venga re-inserito o modificato successivamente: nel caso in esame non c'è possibilità di re-inserire il seed oltre la fase iniziale di setup, poiché durante la simulazione il gestore della classe *Rngs.Java* è *RandomFunction.Java*, dove quest'ultimo è un Singleton per il sistema; dunque, la condizione di non modifica del seed è sicuramente rispettata in ogni run.

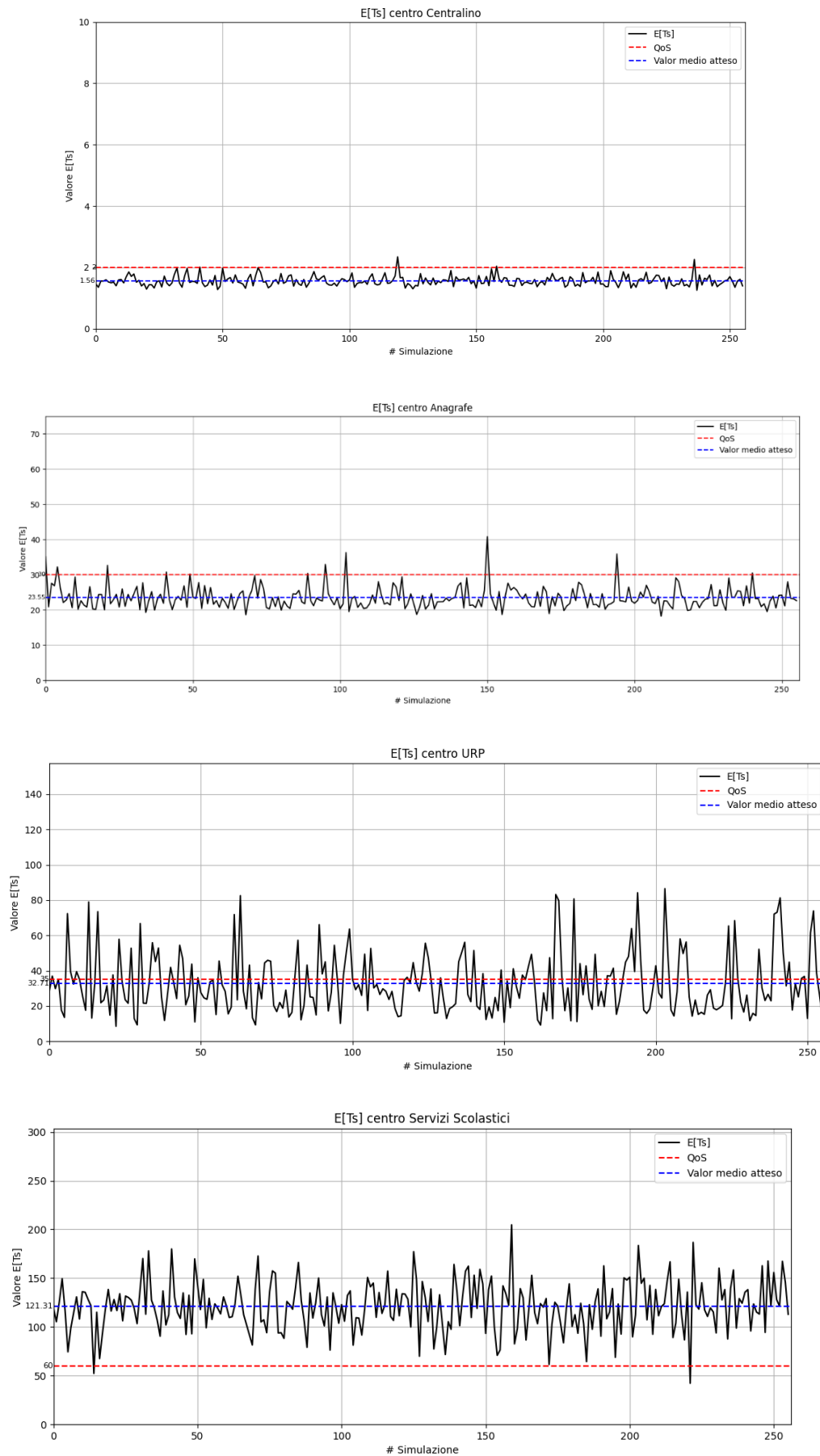
Le simulazioni successive riutilizzano lo stream di valori random dallo stato finale della simulazione precedente: questo ci permette di assumere che le stime generate siano indipendenti e non si sovrappongano. Lo stato dei centri, invece, per ogni simulazione viene re-impostato a condizione iniziale.

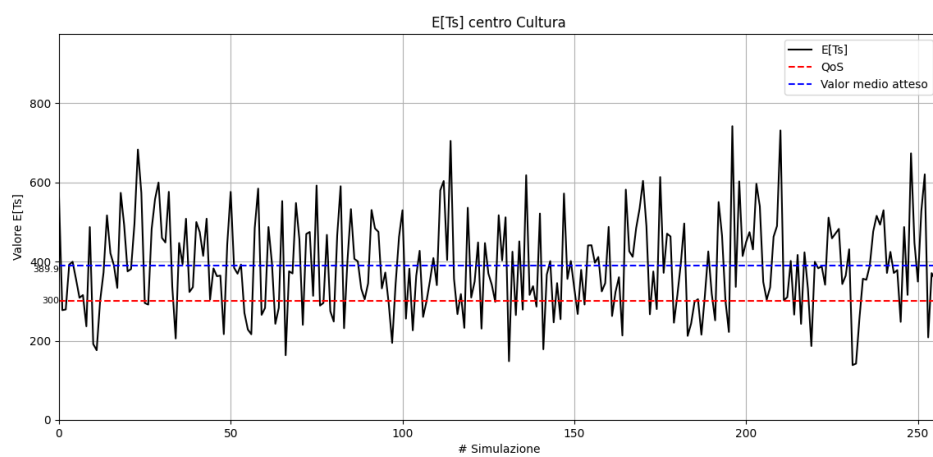
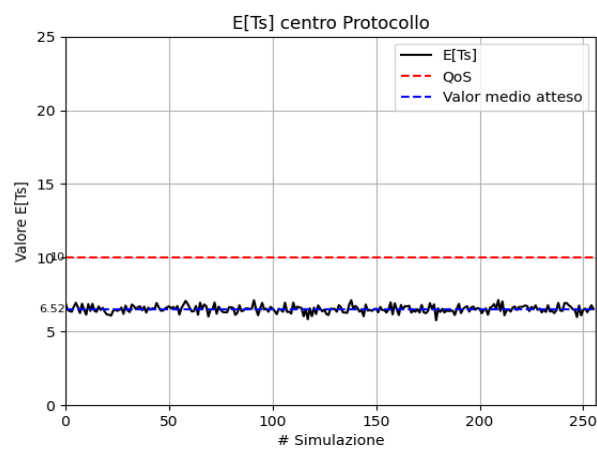
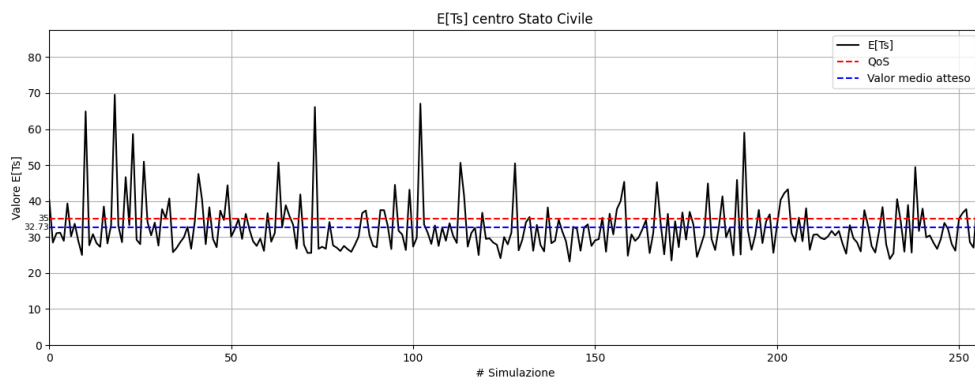
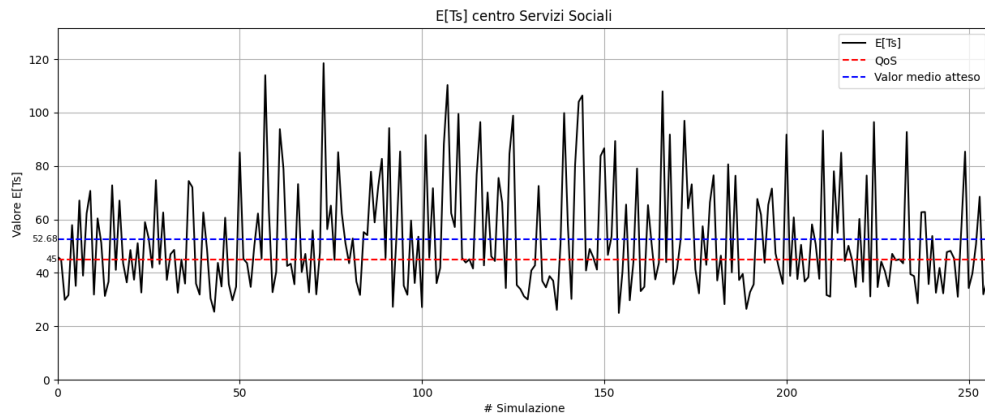
La configurazione dei numeri di server nella simulazione non differisce dalle precedenti simulazioni ma viene riportata di seguito per completezza: {1, 3, 1, 4, 3, 2, 3, 2}.

I valori calcolati dalla *Estimate.Java*, derivanti dalle 256 repliche sono i seguenti:

	ρ	$E(T_s)$	$E(N_s)$
Centralino	0.51 ± 0.00	1.56 ± 0.02	0.79 ± 0.02
Anagrafe	0.62 ± 0.01	23.55 ± 0.40	2.22 ± 0.07
URP	0.89 ± 0.01	32.71 ± 2.13	4.79 ± 0.32
Scolastico	0.98 ± 0.00	121.31 ± 3.16	30.08 ± 0.78
Servizi Sociali	0.88 ± 0.01	52.68 ± 2.51	5.65 ± 0.30
Stato civile	0.66 ± 0.01	32.73 ± 0.92	1.78 ± 0.08
Protocollo	0.23 ± 0.00	6.52 ± 0.03	0.70 ± 0.01
Cultura	0.91 ± 0.01	389.90 ± 14.59	5.89 ± 0.23

Di seguito vengono riportati i grafici dei tempi di risposta di ognuno dei centri del sistema: vengono messi in correlazione grafica i valori medi con i valori dei QoS ricercati.





Come è possibile notare dalle immagini e dai dati raccolti, i centri che non rispettano i QoS sono i “Collo di Bottiglia” evidenziati nel capitolo 8.1 quali: Cultura, Servizi Scolastici e Servizi Sociali. Questi tre centri saranno il punto di partenza per lo studio della versione migliorativa, descritta nel prossimo capitolo. Sono presenti altri due centri, quali URP e Stato Civile che presentano un tempo di risposta molto variabile ma che mediante si attesta comunque ad un valore inferiore dei QoS. I restanti centri non saranno oggetto di studio nella versione migliorativa poiché rispettano i QoS prefissati.

A differenza della simulazione ad orizzonte infinito, la presente simulazione ad orizzonte finito si presta come più attendibile per la studio della versione migliorativa: l’obiettivo di questo studio è il miglioramento dei tempi di risposta della singola giornata lavorativa e non in un lungo periodo.

Da questi valori vengono generate le considerazioni successive per le versioni migliorative.

9 – Versione migliorativa

La versione migliorativa viene realizzata a partire dalle analisi del modello concettuale e delle specifiche applicate al modello base. La simulazione avverrà mediante simulazione ad orizzonte finito e si andranno a modificare i seguenti parametri dei soli centri evidenziati come “Collo di Bottiglia”:

- Tempo di esecuzione;
- Numero di server.

Si andranno, in seguito, ad analizzare le metriche del sistema in risposta alle modifiche apportate.

La modifica sul tempo di esecuzione ipotizza l’attuazione di miglioramenti alle tecniche di lavoro standard, in modo da rispondere alle richieste del cittadino in maniera più efficiente.

La modifica sul numero di server prevede invece l’aumento del numero di server in un centro e il successivo studio della risposta, in modo da capire quale numero di server esatto è in grado di soddisfare i vincoli dei QoS.

Si ipotizza che le differenze sostanziali tra i due miglioramenti precedentemente introdotti siano:

- L’aumento del numero di server è sicuramente realizzabile nel breve periodo, con conseguente aumento notevole dei costi totali.
- Il miglioramento dei tempi di servizio prevede un costo inferiore, a discapito del tempo necessario per studiare tecniche migliorative al processo di lavoro.

9.1 Tempo di esecuzione

In questo paragrafo vengono presentati gli studi effettuati ed i risultati ottenuti andando a variare il tempo di esecuzione dei centri più problematici.

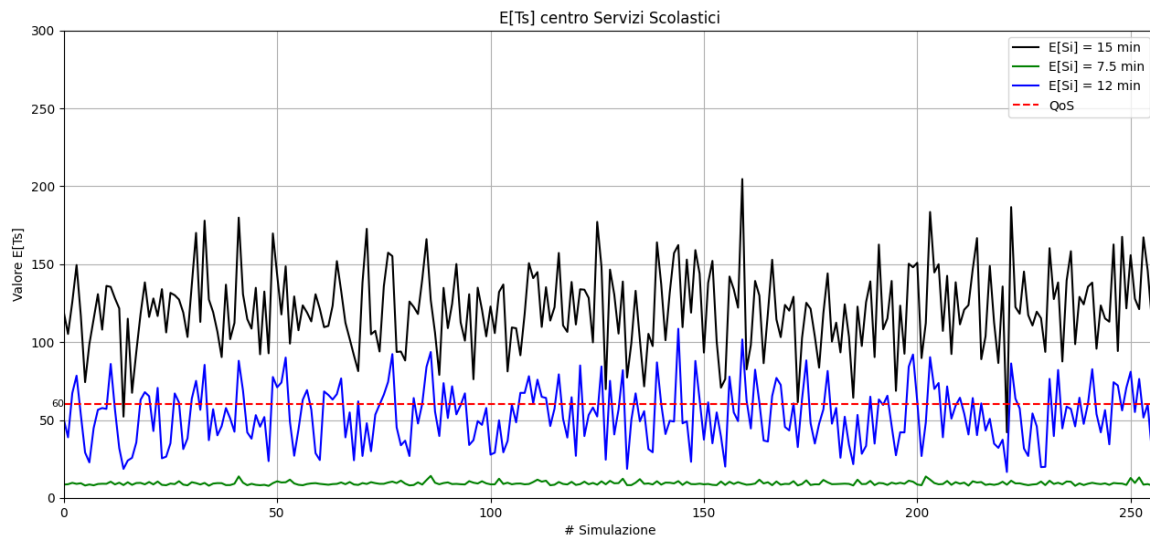
9.1.1 Servizi scolastici

Il centro Servizi scolastici nel modello base del sistema prevede un tempo di servizio $E(S_i) = 15 \text{ minuti}$.

In una prima analisi, si ipotizza di riuscire a dimezzare il tempo di esecuzione del centro, mentre successivamente si è cercato di trovare il valore di $E(S_i)$ tale per cui il vincolo sul tempo di risposta dei QoS venga rispettato (<60 minuti).

	ρ	$E(T_s)$
$E(S_i) = 15$	0.98 ± 0.00	121.31 ± 3.16
$E(S_i) = 7.5$	0.74 ± 0.06	9.38 ± 0.14

$E(S_i) = 12$	0.96 ± 0.00	53.47 ± 2.29
---------------	-----------------	------------------



Dai risultati ottenuti si osserva che:

- Dimezzando il tempo di servizio si ottiene un notevole incremento delle prestazioni, rispettando i vincoli di QoS sia sull'utilizzazione del centro (<80%) sia sul tempo di risposta $E(T_s)$ (<60 minuti).
- Diminuendo il tempo di servizio medio $E(S_i)$ fino a 12 minuti/job (diminuzione del 20%) si nota un incremento delle prestazioni rispettando esclusivamente il vincolo sul tempo di risposta $E(T_s)$ (< 60 minuti).

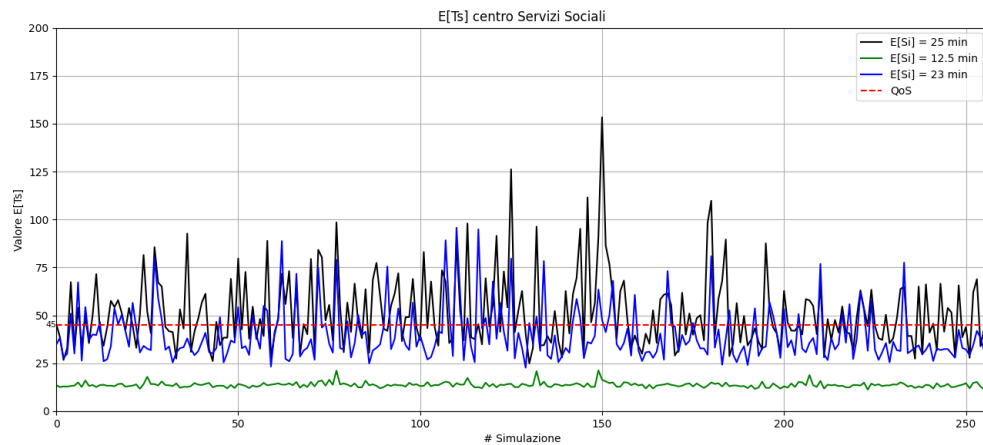
Tra le due soluzioni presentate, la prima è quella migliore, ma con molta probabilità anche molto difficile da implementare. La seconda soluzione presenta il rispetto del vincolo sul tempo di risposta $E(T_s)$, ma non è accettabile poiché l'utilizzazione del nodo rimane eccessivamente elevata.

9.1.2 Servizi sociali

Il centro Servizi Sociali nel modello base del sistema prevede un tempo di servizio $E(S_i) = 25 \text{ minuti}$.

Anche in questo caso, si effettua prima uno studio delle statistiche in caso di tempo di servizio dimezzato, e in seconda analisi si cerca di trovare il valore di $E(S_i)$ tale per cui venga rispettato il vincolo dei QoS sul tempo di risposta $E(T_s)$ (<45 minuti).

	ρ	$E(T_s)$
$E(S_i) = 25$	0.88 ± 0.01	52.68 ± 2.51
$E(S_i) = 12.5$	0.52 ± 0.01	13.73 ± 0.17
$E(S_i) = 23$	0.85 ± 0.01	39.80 ± 1.72



Dai risultati ottenuti si osserva che:

- Dimezzando il tempo di servizio si ottiene un notevole incremento delle prestazioni, poiché le metriche ottenute in questo caso sono molto al di sotto dei vincoli di QoS sia sull'utilizzazione del centro ($<80\%$) sia sul tempo di risposta $E(T_s)$ (<45 minuti).
- Diminuendo il tempo di servizio medio $E(S_i)$ a 23 minuti/job si nota un incremento delle prestazioni rispettando il vincolo sul tempo di risposta $E(T_s)$ (<45 minuti). In questa seconda simulazione anche il valore dell'utilizzazione, pur non essendo al di sotto del vincolo di QoS presenta comunque un valore accettabile.

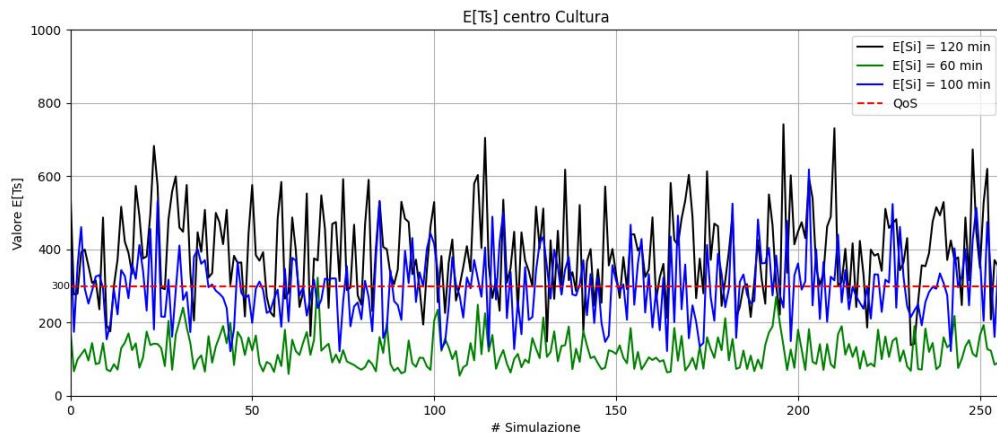
Anche in questo caso tra le due soluzioni presentate, la prima è quella migliore, ma con molta probabilità anche molto difficile da implementare. La seconda soluzione questa volta si ipotizza sia molto facile da raggiungere, in quanto è richiesta una diminuzione sul tempo di servizio medio di soli 2 minuti (pari ad una diminuzione dell'8%).

9.1.3 Cultura

Il centro Cultura nel modello base del sistema prevede un tempo di servizio $E(S_i) = 120$ minuti.

Anche in questo caso, si effettua prima uno studio delle statistiche in caso di tempo di servizio dimezzato, e in seconda analisi si cerca di trovare il valore di $E(S_i)$ tale per cui venga rispettato il vincolo dei QoS sul tempo di risposta $E(T_s)$ (<300 minuti).

	ρ	$E(T_s)$
$E(S_i) = 120$	0.91 ± 0.01	389.90 ± 14.59
$E(S_i) = 60$	0.82 ± 0.01	120.41 ± 5.47
$E(S_i) = 100$	0.90 ± 0.01	298.17 ± 11.27



Dai risultati ottenuti si osserva che:

- Dimezzando il tempo di servizio si ottiene un incremento delle prestazioni, poiché il tempo di risposta del centro in questo caso rispetta il vincolo di QoS (<300 minuti) anche se l'utilizzazione del centro rimane comunque leggermente elevata.
- Diminuendo il tempo di servizio medio $E(S_i)$ a 100 minuti/job si nota un incremento delle prestazioni rispettando di poco vincolo sul tempo di risposta $E(T_s)$ (< 300 minuti). Nonostante il tempo di risposta al di sotto del vincolo richiesto, l'utilizzazione del centro rimane molto alta, e si discosta di poco dal valore ottenuto con i parametri del caso base.

Anche in questo caso tra le due soluzioni presentate, la prima è quella migliore, ma con molta probabilità anche molto difficile da implementare. La seconda soluzione anche in questo caso si ipotizza sia più facile da realizzare (richiesta una diminuzione del tempo di servizio del 16%), ma non migliora del tutto il sistema poiché l'utilizzazione del centro rimane eccessivamente elevata.

9.2 Numero di serventi

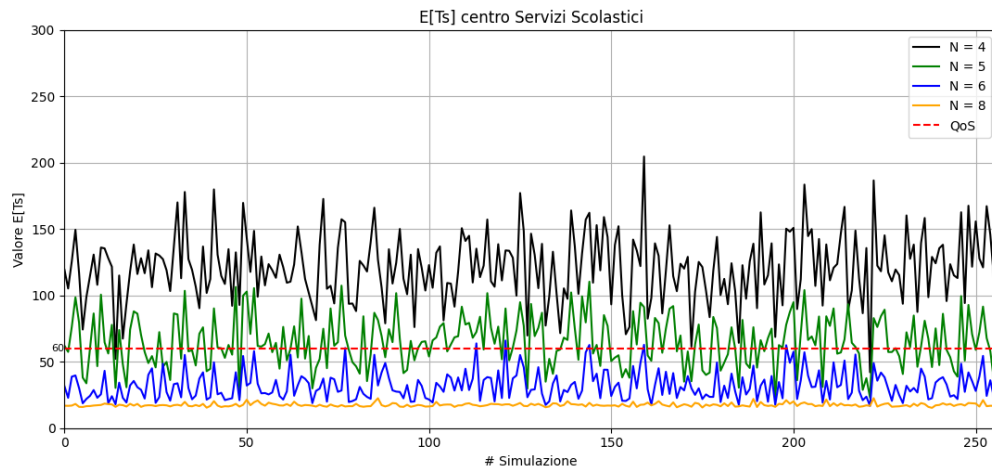
In questo paragrafo vengono presentati gli studi e i risultati ottenuti andando a variare il numero di serventi dei centri più problematici.

9.2.1 Servizi scolastici

Il centro Servizi scolastici nel modello base del sistema prevede un numero di serventi pari a 4.

Nell'analisi che segue, si riporta il variare dei parametri di utilizzazione e tempo di risposta del centro al variare del numero di serventi, fino al raggiungimento del rispetto dei vincoli di QoS.

N. Serventi	ρ	$E(T_s)$
4	0.98 ± 0.00	121.31 ± 3.16
5	0.96 ± 0.00	65.15 ± 2.33
6	0.93 ± 0.00	32.18 ± 1.33
8	0.76 ± 0.01	17.50 ± 0.15



Dai risultati ottenuti si osserva che:

- L'aumento del numero di server di una sola unità è quasi sufficiente a ridurre il tempo di risposta ad un valore al di sotto di quello richiesto dai QoS (<60 minuti). Con l'aumento del numero di server ad un numero pari a 2, si raggiunge ampiamente il valore richiesto per il tempo di risposta, ma il centro risulta ancora presentare un'utilizzazione decisamente elevata.
- Il raggiungimento di entrambi i requisiti di qualità è raggiunto quando il numero di server diventa pari a 8 (raddoppiato rispetto al caso base).

Dalle soluzioni presentate è possibile osservare che è altamente improbabile attuare nel modello reale le modifiche opportune per raggiungere entrambe le richieste dai QoS, poiché questo richiederebbe l'assunzione di 4 ulteriori dipendenti.

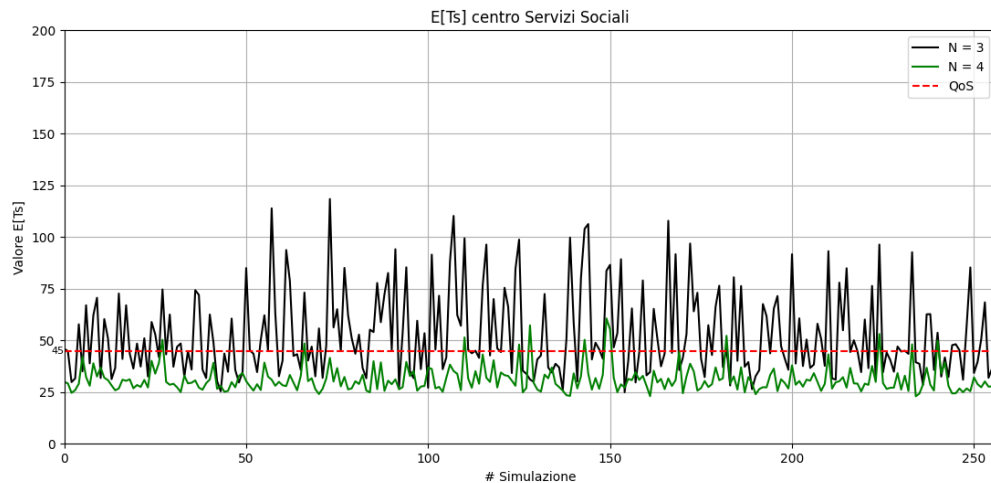
Si ipotizza che la soluzione più corretta per questo centro sia quella di agire su entrambi i parametri, tempo di servizio e numero di server. Questo tipo di soluzione verrà approfondita più avanti nell'elaborato.

9.2.2 Servizi sociali

Il centro Servizi Sociali nel modello base del sistema prevede un numero di server pari a 3.

Nell'analisi che segue, si riporta il variare dei parametri di utilizzazione e tempo di risposta del centro al variare del numero di server, fino al raggiungimento del rispetto dei vincoli di QoS.

N. Server	ρ	$E(T_s)$
3	0.88 ± 0.01	52.68 ± 2.51
4	0.73 ± 0.01	31.25 ± 0.78



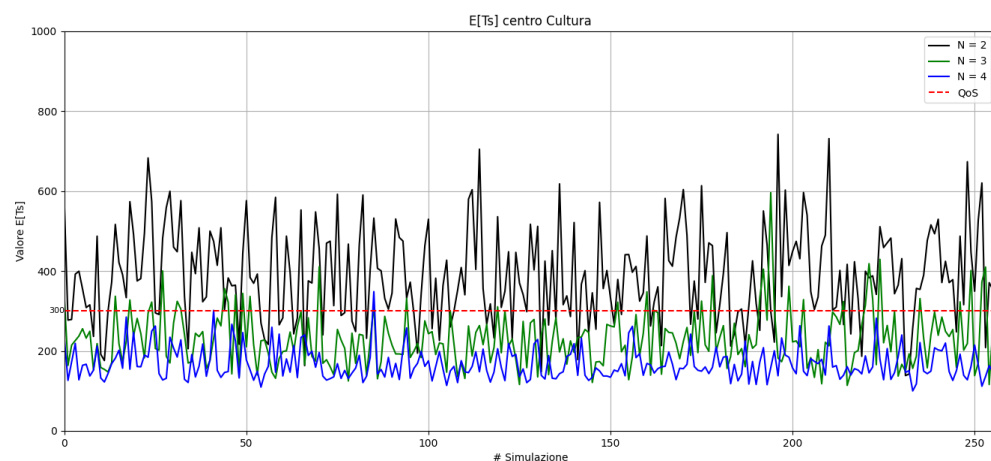
Dai risultati ottenuti si osserva che in questo caso, l'aumento del numero di server di una sola unità è ampiamente sufficiente per portare entrambe le metriche richieste, al di sotto dei valori richiesti dal QoS ($E(T_s) < 45 \text{ minuti}$, $\rho < 80\%$).

9.2.3 Cultura

Il centro Servizi Sociali nel modello base del sistema prevede un numero di server pari a 2.

Nell'analisi che segue, si riporta il variare dei parametri di utilizzazione e tempo di risposta del centro al variare del numero di server, fino al raggiungimento del rispetto dei vincoli di QoS.

N. Server	ρ	$E(T_s)$
2	0.91 ± 0.01	389.90 ± 14.59
3	0.83 ± 0.01	230.09 ± 8.61
4	0.75 ± 0.01	168.25 ± 4.93



Dai risultati ottenuti si osserva che:

- l'aumento del numero di server di una sola unità è sufficiente in questo caso per portare il tempo di risposta del sistema al di sotto del valore richiesto dal QoS ($E(T_s) < 300 \text{ minuti}$) e permette di avere un'utilizzazione quasi accettabile;

- l'aumento del numero di server di due unità è ampiamente sufficiente a portare entrambe le metriche osservate, al di sotto dei valori richiesti dai QoS ($E(T_s) < 300 \text{ minuti}$, $\rho < 80\%$).

Dalle analisi dei risultati è possibile osservare che ipoteticamente la soluzione migliore è quella di aumentare di una unità il numero di server, e agire sul tempo di servizio di questi per raggiungere completamente entrambi i vincoli posti dai QoS. Anche in questo caso, questo tipo di soluzione verrà analizzata più avanti nell'elaborato.

9.3 Versione definitiva

In questo paragrafo vengono illustrate le metriche in analisi del sistema, cercando di combinare i risultati ottenuti precedentemente, agendo sia sul numero di server che sul tempo di elaborazione di ogni server, al fine di suggerire una possibile versione definitiva e realizzabile del sistema che rispetti i QoS.

9.3.1 Servizi scolastici

Il centro Servizi scolastici nel modello base del sistema prevede un numero di server pari a 4, e un tempo di servizio $E(S_i) = 15$. Nel capitolo precedente è stato dimostrato come con l'aumentare del numero di server a 5, il tempo di servizio si attesta intorno ad un valore che si avvicina a quello richiesto dai QoS.

Da queste ipotesi, sono state fatte le seguenti analisi, ipotizzando non si riesca comunque a portare il valore di $E(S)$ al di sotto di 12 minuti:

N. Server	$E(S_i)$	ρ	$E(T_s)$
5	$E(S_i) = 12$	0.90 ± 0.01	22.01 ± 0.93
6	$E(S_i) = 12$	0.78 ± 0.01	14.58 ± 0.21
6	$E(S_i) = 13$	0.82 ± 0.01	16.54 ± 0.34

Dai risultati sopra riportati è possibile notare che aumentando il numero di server di una sola unità e diminuendo il tempo medio di servizio a 12 minuti, non permette comunque al sistema di raggiungere i valori imposti dai QoS.

La soluzione ottimale è dunque la seguente:

- N. Server = 6
- $E(S_i) = 12 \text{ minuti}$

9.3.2 Servizi sociali

Il centro Servizi Sociali nel modello base del sistema prevede un numero di server pari a 3, e un tempo di servizio $E(S_i) = 25$ minuti. Nei capitoli precedenti è stato dimostrato che agire esclusivamente sui tempi di servizio medi, permette di attestare il valore del tempo di risposta al di sotto di quello richiesto dai QoS e permette di avere una utilizzazione pari all'85%; invece, con l'aumentare del numero di server a 4, entrambe le metriche in analisi, soddisfano i requisiti di QoS.

La soluzione ottimale in questo caso non è unica, ma è necessario affrontare una analisi più approfondita sui costi:

- Diminuire esclusivamente il tempo di servizio richiesto dell'8%, permette di soddisfare parzialmente i QoS, senza andare ad intaccare eccessivamente i costi necessari al miglioramento.

- Aumentare il numero di serventi di una unità, permette di rispettare entrambi i requisiti di QoS, ma con una spesa da affrontare per il centro comunale sicuramente maggiore.

9.3.3 Cultura

Il centro Cultura nel modello base del sistema prevede un numero di serventi pari a 2 e un tempo di servizio $E(S_i) = 120$ minuti. Nel capitolo precedente è stato dimostrato come con l'aumentare del numero di serventi a 3, il tempo di servizio rispetta ampiamente il valore imposto dal QoS e l'utilizzazione del centro si attesta intorno a valori accettabili (83%).

La soluzione ottimale, che permette di soddisfare entrambi i requisiti, è dunque la seguente:

- N. Serventi = 3
- $E(S) = 120$ minuti

10 – Conclusioni

Lo studio effettuato ha permesso di mettere in luce alcune delle criticità presenti, in particolare, all'interno degli uffici rivolti ai cittadini del Comune di Lanciano.

Il problema maggiore sembra ricollegabile alla mancanza di organico all'interno di alcuni sportelli, con un conseguente sovraccarico del lavoro. Va tenuto conto però delle difficoltà di assunzione di nuovo organico che si riscontrano in tutti i servizi pubblici presenti sul territorio nazionale.

Un doveroso ringraziamento va al Comune di Lanciano e ai suoi dipendenti per la possibilità offerta e per i dati forniti.