

# Analisi delle Prestazioni di un Autofficina Meccanica

Progetto del corso Performance Modeling of Computer Systems and Networks

Tummolo Gabriele Matricola 0283629

## Sommario

<b>Introduzione</b> .....	1
<b>Modello Concettuale</b> .....	2
<b>Modello delle Specifiche</b> .....	3
<b>Modello computazionale</b> .....	3
<b>Calcolo valori teorici</b> .....	3
<b>Verifica</b> .....	4
<b>Validazione</b> .....	5
<b>Progettazione della simulazione &amp; simulazione</b> .....	5
<b>Analisi Transiente</b> .....	7
<b>Analisi Stazionaria</b> .....	8
<b>Conclusione</b> .....	9

## Introduzione

Questa relazione riassume il lavoro svolto per il progetto del corso di Performance Modelling of Computer Systems and Networks, tenuto dalla prof.ssa Vittoria De Nitto Personè al primo anno del Corso di Laurea Magistrale in Ingegneria Informatica presso l'Università Roma Tor Vergata. Il progetto consiste in un'analisi del tempo di servizio di un'autofficina meccanica.

Per lo sviluppo del progetto sono stati seguiti i passi dell'algoritmo 1.1.1 e 1.1.2 del libro di testo *Discrete-Event Simulation: A First Course*.

Un' autofficina offre servizi di riparazione/manutenzione per sole automobili. La riparazione/manutenzione di un veicolo può essere realizzata mediante l'ausilio di un Ponte, che è una struttura grazie alla quale è possibile sollevare l'automobile per procedere con la riparazione/manutenzione nel modo più efficiente garantendo maggiore sicurezza al meccanico.

Nelle autofficine, in generale, ci sono più ponti che danno la possibilità di servire più clienti contemporaneamente. Inoltre, ci possono essere 2 tipologie di clienti:

- Clienti che conoscono già il problema del veicolo (Tagliando, ovvero sostituzione olio e filtri), che fanno riferimento alle manutenzioni ordinarie
- Clienti che non conoscono il problema del veicolo e quindi questo deve essere prima ispezionato dal meccanico per individuare il problema

In questo tipo di problema ci sono numerose variabili da tenere in considerazione, in quanto una semplice manutenzione ordinaria, in termini di tempo di servizio, può variare da veicolo a veicolo.

Il flusso di automobili in ingresso nell'autofficina può variare anche dal periodo (in inverno ci sono meno automobili & in estate più automobili).

Quindi il modello preso in considerazione fa riferimento al periodo estivo (flusso maggiore).

L'obiettivo dello studio è di trovare il numero minimo di ponti modo tale da avere un tempo medio di servizio inferiore ad un certo valore. Quindi considerando che ogni ponte ha un costo e ad ogni ponte viene assegnato un meccanico specializzato, individuare il numero di ponti che mi permette di avere un costo minimo e che garantisce il tempo di servizio che rispetti il QoS.

In altre parole, vogliamo individuare il numero di ponti minimo per cui il tempo di risposta è inferiore a 12 ore, che sono equivalenti a un giorno e mezzo. Questo perché gli orari che vengono osservati dalle officine sono i seguenti:

- 8:30 - 12:30 Mattina
- 14:30 – 18:30 Pomeriggio

## Modello Concettuale

Il modello concettuale riportato nella figura 1:

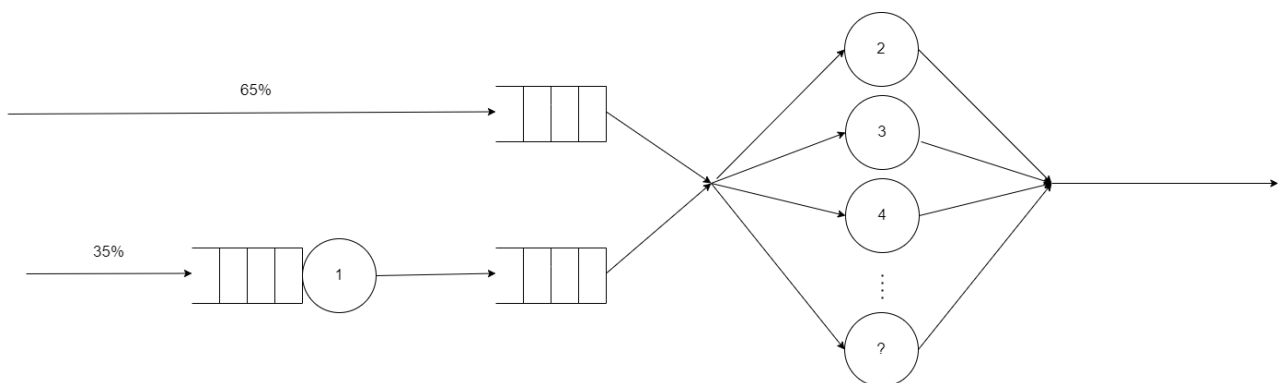


Figura 1 Modell concettuale Autofficina

Il servente 1 rappresenta il servizio di Diagnosi, ovvero il servizio che permette al meccanico di individuare il problema del veicolo per poi successivamente procedere con la riparazione.

Gli altri serventi rappresentano i ponti all'interno della struttura, dove viene fatta la riparazione effettiva del veicolo. Quindi l'autofficina è stata modellata come un Multi-server con 2 code di priorità (Priorità astratta).

Il flusso totale di ingresso nel sistema viene ripartito per il 65% nella prima coda e 35% nella seconda coda, queste percentuali sono state calcolate tramite sondaggi.

Ogni servente (Ponte o Diagnosi) può trovarsi in due differenti stati:

- Libero
- Occupato

Per ogni coda è stata utilizzata una variabile che permette di verificare il numero di veicoli in coda, per la specifica coda, considerando anche il veicolo in servizio (se la variabile ha un valore maggiore o uguale a 1 vuol dire che c'è un veicolo in servizio).

## Modello delle Specifiche

Per la costruzione del modello delle specifiche sono stati utilizzati i seguenti valori:

- Tempo di inter-arrivo Esponenziale con parametro  $\lambda = 0.5$  automobili/ora
- Tempo di servizio (Ponte) Esponenziale con parametro  $\mu_p = 0.2$  automobili/ora
- Tempo di servizio (Diagnosi) Esponenziale con parametro  $\mu_d = 1.66$  automobili/ora

Questi valori come nel caso precedente sono stati individuati tramite sondaggi; infatti, in un periodo di maggiore afflusso di veicoli (periodo estivo) il numero di veicoli che arrivano in una giornata lavorativa è di circa 4 automobili ogni 8 ore; mentre rispettivamente per effettuare una diagnosi con gli strumenti ad oggi esistenti ci vogliono circa 36 minuti, mentre per la riparazione di un'automobile ci vogliono circa 5 ore.

Per questo abbiamo i seguenti valori dei tempi di servizio:

- $E(S_p) = 5$  ore
- $E(S_d) = 36$  minuti

Le singole code adottano una disciplina di servizio FIFO.

Le variabili di stato sono le seguenti:

- $Queue[i]$ , indica il numero di veicoli in coda, abbiamo 3 code
  - Coda Diagnosi
  - Coda Automobili privilegiate
  - Coda Automobili non privilegiate
- $N\_Ponti$  che rappresenta il numero di ponti in autofficina

## Modello computazionale

Il modello in questione fa riferimento al codice scritto e utilizzato per effettuare la simulazione, si tratta di un codice scritto in linguaggio C.

Dopo aver compilato il progetto con il comando *make all* è possibile avviare la simulazione tramite il comando *./Autofficina.o*.

Nella prima fase della simulazione viene chiesto di inserire il seed, questo permette di garantire la riproducibilità della simulazione e quindi avere una maggiore variabilità dei risultati.

Gli altri parametri quali ad esempio *LAMBDA* e *MU* non possono essere modificati a run-time, ma bisogna modificare il file e ricompilare nuovamente per modificare questi valori.

## Calcolo valori teorici

Per confrontare i valori della simulazione con quelli prodotti dal modello teorico sono state utilizzate le seguenti formule:

$$E(S_p) = \frac{1}{\mu_p * N_{ponti}}$$

$$E(S_{p_i}) = \frac{1}{\mu_p}$$

$$E(S_d) = \frac{1}{\mu_d}$$

$$\rho = \frac{\lambda}{\mu_p}$$

$$\rho_d = \frac{0.35 * \lambda}{\mu_d}$$

$$\rho_1 = \rho * 0.65$$

$$\rho_2 = \rho * 0.35$$

$$P_{Q1} = \text{Erlang}(\rho_1) = \left[ \sum_{i=0}^{N_{\text{ponti}}-1} \frac{(N_{\text{ponti}} * \rho_1)^i}{i!} + \frac{(N_{\text{ponti}} * \rho_1)^{N_{\text{ponti}}}}{N_{\text{ponti}}! * (1 - \rho_1)} \right]^{-1} * \left( \frac{(N_{\text{ponti}} * \rho_1)^{N_{\text{ponti}}}}{N_{\text{ponti}}! * (1 - \rho_1)} \right)$$

$$P_Q = \text{Erlang}(\rho) = \left[ \sum_{i=0}^{N_{\text{ponti}}-1} \frac{(N_{\text{ponti}} * \rho)^i}{i!} + \frac{(N_{\text{ponti}} * \rho)^{N_{\text{ponti}}}}{N_{\text{ponti}}! * (1 - \rho)} \right]^{-1} * \left( \frac{(N_{\text{ponti}} * \rho)^{N_{\text{ponti}}}}{N_{\text{ponti}}! * (1 - \rho)} \right)$$

$$E(T_{q\_Diagnosi}) = \frac{\rho d * E(S_d)}{(1 - \rho d)}$$

$$E(T_{s\_Diagnosi}) = E(T_{q\_Diagnosi}) + E(S_d)$$

$$E(T_{q\_Multiserver}) = 0.65 * \frac{PQ1 * E(Sp)}{(1 - \rho_1)} + 0.35 * \frac{PQ * E(Sp)}{(1 - \rho)(1 - \rho_1)}$$

$$E(T_{s\_Multiserver}) = E(T_{q\_Multiserver}) + E(S_{p\_i})$$

La formula per il calcolo del tempo di risposta del sistema è stata calcolata in questo modo in quanto il modello rappresenta una rete di Jackson.

$$E(T_s) = 0.35 * E(T_{s\_Diagnosi}) + E(T_{s\_Multiserver})$$

Per il calcolo di questi valori e della metrica  $E(T_s)$  è stato scritto il file *ValoriTeorici.c* che permette di calcolare velocemente il valore corretto.

## Verifica

Questo passo è molto importante in quanto permette di valutare la correttezza del simulatore, in particolare sono stati calcolati i valori delle metriche e confrontati con i valori teorici. In particolare, è stato avviato il file della simulazione con un tempo di esecuzione molto alto (30000 ore) impostando un punto di inizio casuale, è stato utilizzato il seed = 12345678.

Per la verifica il parametro dello studio (Numero di ponti) è stato fissato a un valore pari a 3.

I risultati del simulatore sono stati confrontati con i valori teorici calcolati utilizzando le formule riportate nel paragrafo *Calcolo valori teorici*, e vengono rappresentati nella seguente tabella:

Statistica	Valore Simulazione	Valore Teorico	Percentuale differenza valore teorico
<b>E(S<sub>Diagnosi</sub>)</b>	0,601971	0,602410	0,1%
<b>E(T<sub>q_Diagnosi</sub>)</b>	0,070956	0,070991	0,1%
<b>E(T<sub>s_Diagnosi</sub>)</b>	0,672927	0,673401	0,2%
<b>E(S<sub>Multiserver</sub>)</b>	4,995270	5,000000	0,1%
<b>E(T<sub>q_Multiserver</sub>)</b>	6,891227	6,032495	14%
<b>E(T<sub>s_Multiserver</sub>)</b>	11,886498	11,032495	7%
<b>E(T<sub>s</sub>)</b>	12,121826	11,268185	7%

Tutti i valori presenti in tabella hanno come unità di misura Ore.

Come si può notare dalla tabella tutti i valori differiscono dal valore teorico al massimo di un valore del 7% a differenza del tempo di attesa in coda per il multiserver che ha percentuale di errore del 14%.

Ma il parametro importante, ovvero il tempo medio di servizio (ultimo elemento della tabella) ha una percentuale di errore del 7%.

## Validazione

Questo passo permette di verificare se il simulatore è rappresentativo del sistema reale.

È stato eseguito più volte il simulatore variando i vari parametri quali *LAMBDA*, *MU* e *N\_Ponti*, verificando che siano rispettate le seguenti condizioni:

- Con l'aumento di *LAMBDA* il tempo di risposta del sistema aumenta
- Con l'aumento di *MU* il tempo di risposta del sistema diminuisce
- Aumentando *N\_Ponti* il tempo di risposta del sistema diminuisce

Per tutte le simulazioni è stato fissato il seed a 12345678.

La tabella rappresenta i risultati ottenuti facendo variare un parametro per volta.

LAMBDA	MU	Numero Ponti	E(TS)
0,5	0,2	3	12,121826
0,6	0,2	3	851,344492
0,5	0,3	3	4,306447
0,5	0,4	3	2,953600
0,5	0,2	4	6,286665
0,5	0,2	5	5,488253

Come si evince dalla seguente tabella le condizioni sopra vengono rispettate; quindi, il simulatore è conforme con un sistema reale.

## Progettazione della simulazione & simulazione

Per individuare il numero minimo di ponti che permette di soddisfare i QoS è stato utilizzato il simulatore variando il parametro *N\_Ponti*.

Per la simulazione sono stati fissati i valori *LAMBDA*, *MU* e *STOP* che indica il tempo di fine simulazione.

Il valore di *STOP* è stato fissato al valore 30000.

Ogni simulazione con un determinato valore di *N\_Ponti* è stato eseguito 8 volte con 8 diversi seed.

Si è iniziati a considerare un numero di ponti pari a 3 in quanto con valori minori di 3 l'utilizzazione è maggiore di 1 ( $p > 1$ ).

I risultati sono stati rappresentati nella seguente figura e nella tabella successiva:

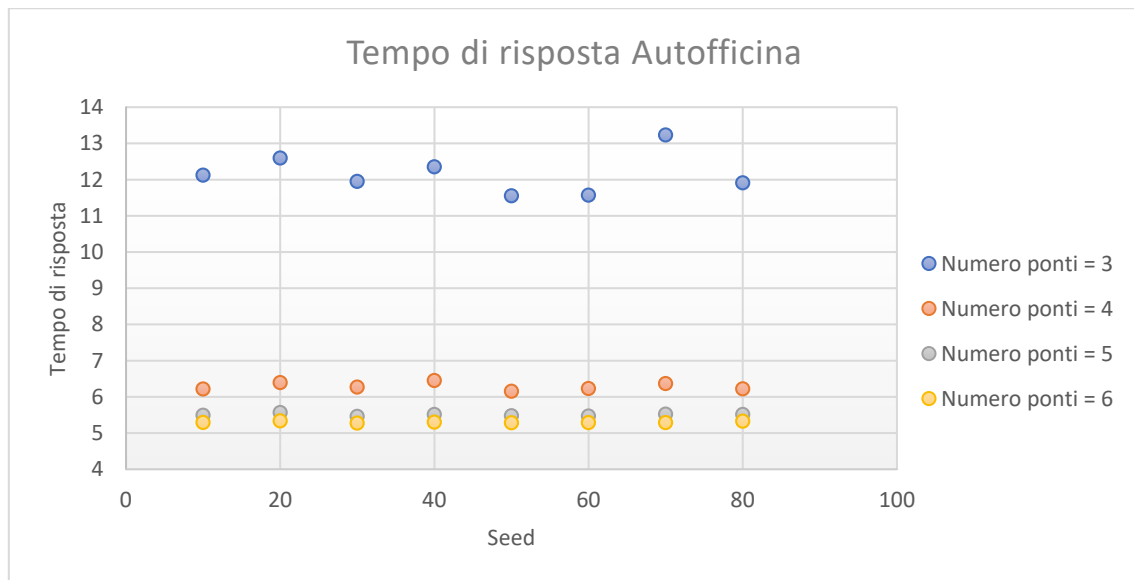


Figura 2 il grafico rappresenta il valore di  $E(T_s)$  in corrispondenza di 8 diverse simulazioni con il numero di ponti da 3 a 6

Seed	Numero Ponti	$E(T_s)$
10	3	12,1207
20	3	12,5916
30	3	11,9463
40	3	12,3512
50	3	11,5493
60	3	11,5687
70	3	13,2283
80	3	11,9071
10	4	6,2132
20	4	6,3906
30	4	6,2672
40	4	6,4468
50	4	6,1514
60	4	6,2262
70	4	6,3659
80	4	6,2186
10	5	5,4868
20	5	5,5631
30	5	5,4557
40	5	5,5089
50	5	5,4768
60	5	5,4664
70	5	5,5177
80	5	5,5091
10	6	5,2898
20	6	5,3339
30	6	5,2716

40	6	5,2971
50	6	5,2814
60	6	5,2865
70	6	5,2873
80	6	5,3243

Come si nota dalla figura 2 passando da 3 a 4 ponti dimezziamo il tempo di risposta passando da circa 12 ore a 6 ore, definendo così il valore minimo di ponti necessari per avere un tempo di risposta minore di 12 ore.

## Analisi Transiente

Per effettuare lo studio dell'analisi transiente è stato utilizzato il meccanismo delle repliche. Nello specifico dopo aver fissato il seed = 12345678 è stato avviato il simulatore utilizzando diversi tempi di esecuzione, in particolare i tempi variano da 160 a 30000 ore.

Il file utilizzato per effettuare questo tipo di analisi è *Transiente.c* che è identico al simulatore ma permette di effettuare più simulazioni con un solo run. I risultati delle simulazioni vengono salvati su file specifici che successivamente sono stati utilizzati per creare gli intervalli di confidenza al 95%. Per ogni tempo sono state utilizzate 50 repliche, ogni replica produce una stima di  $E(T_s)$  indipendente dalle altre.

Per mantenere l'indipendenza tra le diverse repliche sono state inizializzate tutte le variabili ai valori iniziali, questo è stato fatto ad ogni replica.

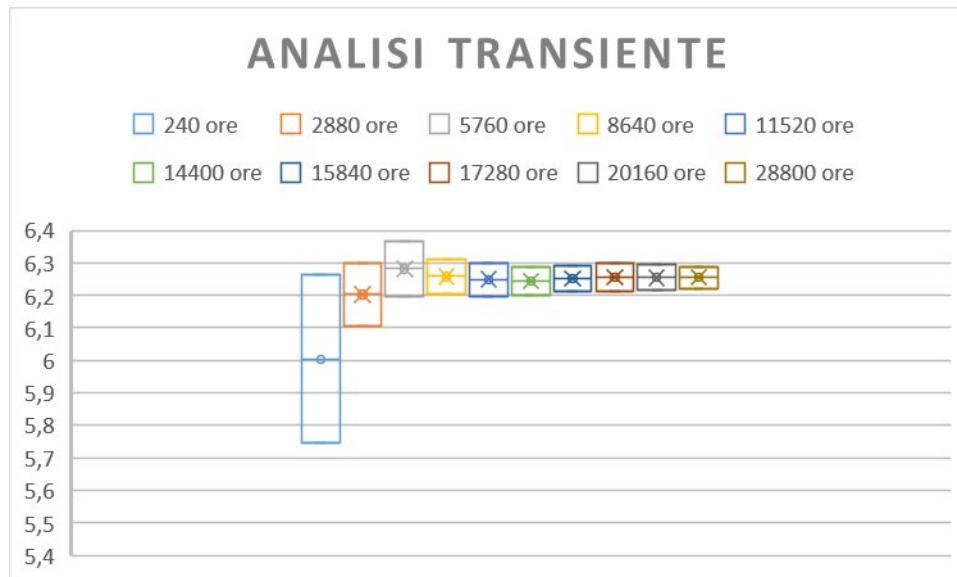


Figura 3 grafico dell'analisi transiente che mostra gli intervalli di confidenza al 95% per i diversi tempi di esecuzione considerati

Come si può notare dalla figura il sistema inizia a stabilizzarsi dopo circa 11520 ore che corrispondono a circa 3 anni; infatti, con valori maggiori di 11520 il tempo di risposta del sistema rimane stabile intorno al valore 6,25 ore che è molto inferiore alle 12 ore.

Sono stati riportati in tabella anche i valori degli intervalli di confidenza.

Tempo simulazione (ore)	Valore medio	Ampiezza
<b>240</b>	6,00387	0,259217
<b>2880</b>	6,202912	0,097803
<b>5760</b>	6,281832	0,085153
<b>8640</b>	6,258102	0,054944
<b>11520</b>	6,249373	0,051995
<b>14400</b>	6,244062	0,043642
<b>15840</b>	6,25224	0,040385
<b>17280</b>	6,25626	0,042694
<b>20160</b>	6,254647	0,039023
<b>28800</b>	6,25535	0,034089

Successivamente si è proceduto con lo studio dello stato stazionario utilizzando il metodo dei Batch Means, con il fine ultimo di valutare il comportamento del sistema quando si raggiunge la stazionarietà eliminando il problema del bias dovuto alle condizioni iniziali.

## Analisi Stazionaria

Come detto alla fine del capitolo precedente l'analisi stazionaria è stata affrontata utilizzando il metodo dei Batch Means. È molto importante scegliere la dimensione dei Batch (B) e il numero di Batch (K), per fare ciò sono state osservate le seguenti linee guida:

- $B > 1$
- $K \geq 32$
- $N = B * K$  (identifica il numero di Veicoli da processare)
- Banks, Carson, Nelson, and Nicol (2001, page 438) scegliere B in modo tale che l'autocorrelazione tra i Batch Means sia minore di 0.2

Come visto nell'analisi transiente si raggiunge uno stato stazionario a circa 11500 ore nel quale vengono processati circa 7500 veicoli, ottenendo così i seguenti valori:

- $N = 7500$
- $K = 64$
- $B = 117$

Si può notare che le prime 2 linee guida sono rispettate, per verificare la quarta linea guida è stato utilizzato il file *acs.c* che ha permesso di ottenere l'autocorrelazione campionaria tra i batch means; il valore di autocorrelazione ottenuto è 0.015 che è inferiore al valore limite 0.2.

Il file utilizzato per effettuare l'analisi è *Stazionaria.c*.

Per effettuare l'analisi come nei casi precedenti è stato fissato il seed a 12345678 e sono stati calcolati 10 diversi intervalli di confidenza al 95% modificando gli stream, i risultati sono stati graficati nella figura successiva.



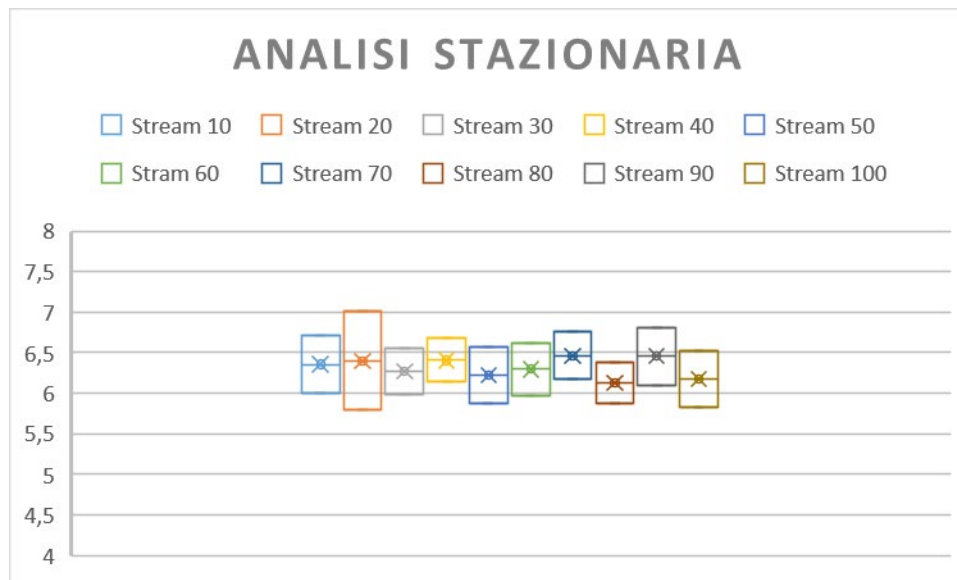


Figura 4 grafico dell'analisi stazionaria con i diversi intervalli di confidenza al 95% in riferimento a differenti stream

Dalla figura 4 si può notare che gli intervalli di confidenza toccano tutti il valore stimato dall'analisi transiente che è circa 6,25. Questo si evince in maniera più evidente dalla tabella sottostante che racchiude tutti i valori degli intervalli calcolati.

Stream	Valore medio	Ampiezza
10	6,356567	0,355094
20	6,400743	0,610157
30	6,269928	0,288238
40	6,406791	0,267957
50	6,222796	0,354196
60	6,296266	0,319101
70	6,462108	0,290707
80	6,127896	0,258274
90	6,456567	0,355094
100	6,173004	0,350011

## Conclusione

L'obiettivo ultimo del progetto era di individuare il numero minimo di ponti in un'autofficina meccanica per avere tempo di risposta inferiore alle 12 ore, il numero minimo permette di avere anche la spesa minima per raggiungere questo obiettivo. La spesa comprende l'acquisto dei ponti (circa 2000 € ogni ponte) più gli stipendi dei meccanici specializzati per ogni ponte (circa 1500 € al mese).

La soluzione proposta prevede di utilizzare 4 ponti affrontando così una spesa di 8000 € più le spese mensili per i meccanici, questo fa sì di garantire al cliente di avere l'auto riparata in meno di 12 che è il limite massimo entro cui gli intervistati hanno espresso i loro bisogni.