模擬與統計 HW6

N26120870 電機所碩一 林耕澤

Homework: Write a Simulation for M/M/1 Queueing System

- Input:
  - Num. of Server (1)
  - Policy of Queue (FIFQ)
  - Customer arrival (Poisson process)
  - Customer departure
  - Service time (Exponential distribution)

- Discuss Output Can utilization go to 1? 100% busy?
  - Server utilization (lambda/mu)
  - Waiting time (expected value, distribution)

    Why plot the distribution of waiting time, but not the histogram?

6-1 模擬說明:

我將模擬 M/M/1 Queueing System 的三種情況:

- λ < u (arrival rate < service rate): 店員服務速度大於來客速度。
- λ = u (arrival rate = service rate): 店員服務速度等於來客速度。
- λ > u (arrival rate > service rate): 店員服務速度慢於來客速度。

6-1 程式碼:

```
def exp_rv_generator(Lambda):
    U = random.random()
    X = - np.log(U) / Lambda
    return X
```

Fig. 1 exponential random variable

```python
def MM1_model_fixed_time_interval(mean_service_rate, mean_arrivel_rate, end_time):
    '''固定時間區間'''
    arrival_time = []
    service_time = []
    waiting_time = []
    departure_time = []
    T = end_time

    i = 0
    arrt = exp_rv_generator(mean_arrivel_rate)
    if arrt > T:
        return arrival_time, service_time, waiting_time, departure_time, 0
    arrival_time.append(arrt)
    waiting_time.append(0)
    service_time.append(exp_rv_generator(mean_service_rate))
    departure_time.append(service_time[0] + waiting_time[0] + arrival_time[0])
    i += 1
    while arrival_time[i - 1] < T:
        arrt = exp_rv_generator(mean_arrivel_rate)
        if arrt + arrival_time[i - 1] >= T:
            return arrival_time, service_time, waiting_time, departure_time, i + 1
        arrival_time.append(arrt + arrival_time[i - 1])

        if departure_time[i - 1] - arrival_time[i] > 0:
            waiting_time.append(departure_time[i - 1] - arrival_time[i])
        else:
            waiting_time.append(0)
        service_time.append(exp_rv_generator(mean_service_rate))
        departure_time.append(arrival_time[i] + service_time[i] + waiting_time[i])
        i += 1
```

Fig. 2 M/M/1 模擬函式(固定結束時間版本)

```python
def MM1_model_fixed_number_of_cumstors(mean_service_rate, mean_arrivel_rate, num):
    '''固定來可客人數'''
    arrival_time = []
    service_time = []
    waiting_time = []
    departure_time = []

    arrival_time.append( exp_rv_generator(mean_arrivel_rate))
    service_time.append( exp_rv_generator(mean_service_rate))
    waiting_time.append(0)
    departure_time.append(arrival_time[0] + service_time[0])
    for i in range(1, num):
        arrival_time.append( arrival_time[i - 1] + exp_rv_generator(mean_arrivel_rate))
        service_time.append( exp_rv_generator(mean_service_rate))

    for i in range(1, num):
        wait_temp = (arrival_time[i - 1] + service_time[i - 1] + waiting_time[i - 1]) - arrival_time[i]
        if wait_temp >= 0:
            waiting_time.append(wait_temp)
        else:
            waiting_time.append(0)

        departure_time.append(arrival_time[i] + service_time[i] + waiting_time[i])

    return arrival_time, service_time, waiting_time, departure_time
```

Fig. 3 M/M/1 模擬函式(固定結束人數版本)

```
arrival_time, service_time, waiting_time, departure_time, i = MM1_model_fixed_time_interval(10,5,100)
arrival_time2, service_time2, waiting_time2, departure_time2, i2 = MM1_model_fixed_time_interval(10,10,100)
arrival_time3, service_time3, waiting_time3, departure_time3, i3 = MM1_model_fixed_time_interval(10,20,100)

#service rate > arrival rate
plt.plot(waiting_time, label ="utilization"+ str(np.sum(service_time)/departure_time[-1]))
plt.legend()
plt.xlabel('cumster #')
plt.ylabel('waiting time')
plt.title('service_rate = 10, arrival_rate = 5, T = 100')
plt.show()
print(np.sum(service_time)/departure_time[-1], i)

#service rate = arrival rate
plt.plot(waiting_time2, label ="utilization"+ str(np.sum(service_time2)/departure_time2[-1]))
plt.legend()
plt.xlabel('cumster #')
plt.ylabel('waiting time')
plt.title('service_rate = 10, arrival_rate = 10, T = 100')
plt.show()
print(np.sum(service_time2)/departure_time2[-1], i2)

#service rate < arrival rate
plt.plot(waiting_time3, label ="utilization"+ str(np.sum(service_time3)/departure_time3[-1]))
plt.legend()
plt.xlabel('cumster #')
plt.ylabel('waiting time')
plt.title('service_rate = 10, arrival_rate = 20, T = 100')
plt.show()
print(np.sum(service_time3)/departure_time3[-1], i3)
```

Fig. 4 模擬函式輸出以及製圖

6-1 程式碼 flow chart(固定結束時間版本)

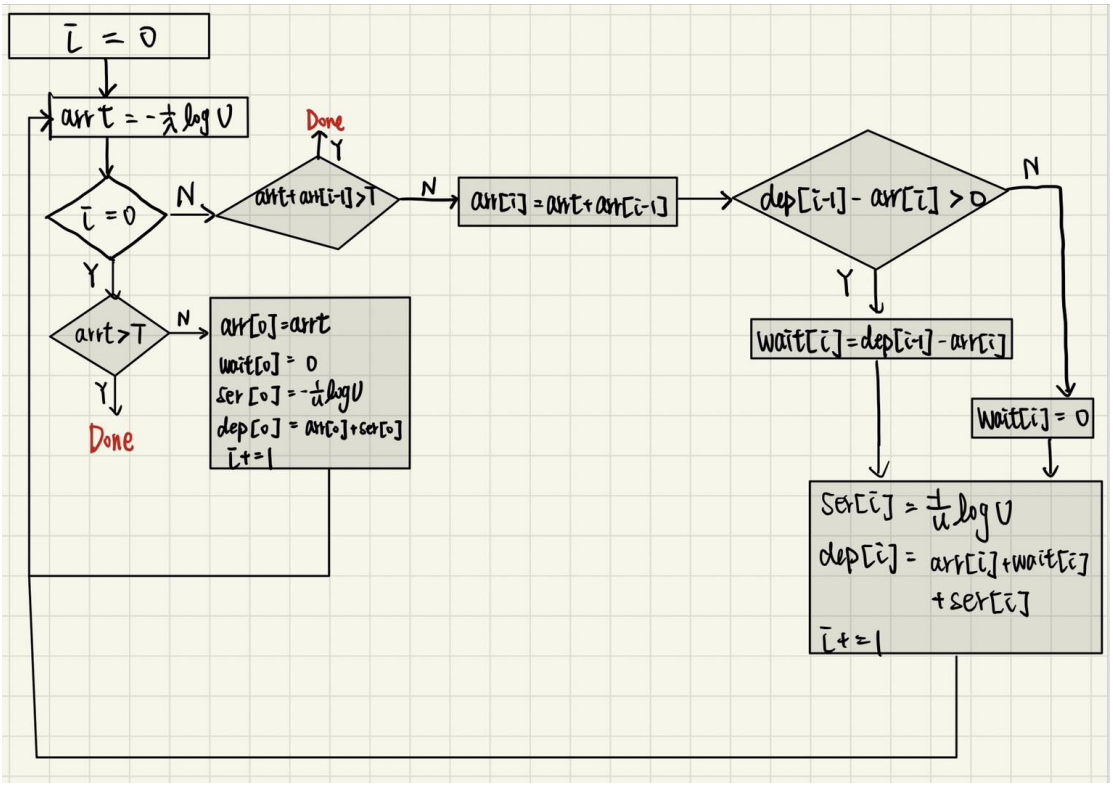

Fig. 5 flow chart ,arr = arrival_time、wait = waiting_time、ser = service_time、dep = departure_time

6-1 模擬結果:

- λ < u (arrival rate < service rate):

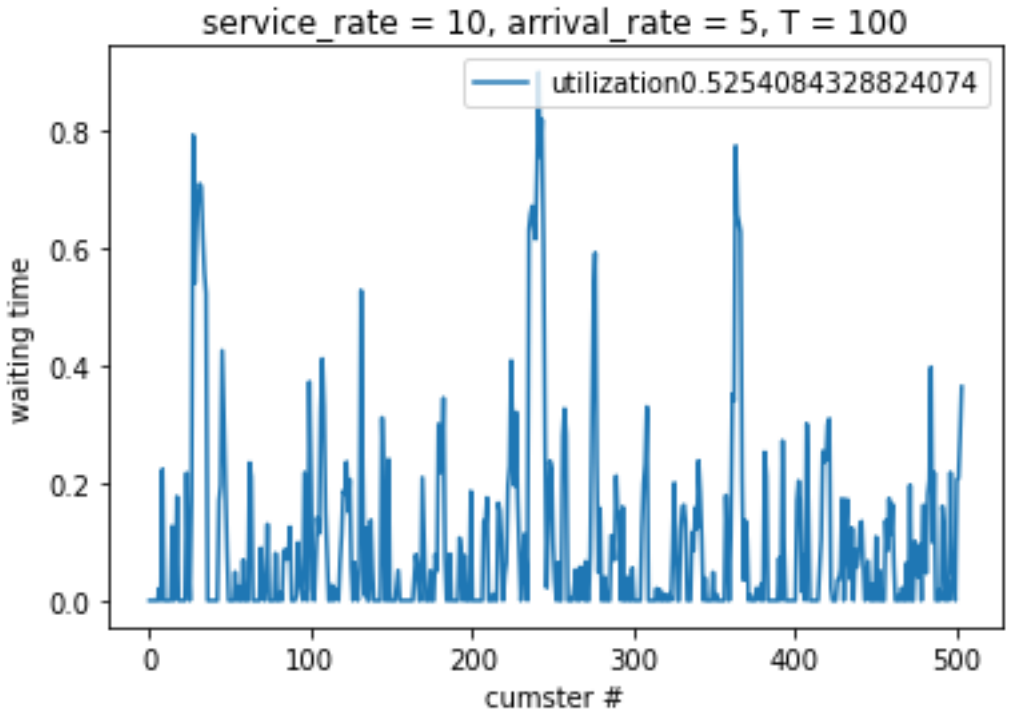

Fig. 6 λ < u

- λ = u (arrival rate = service rate):

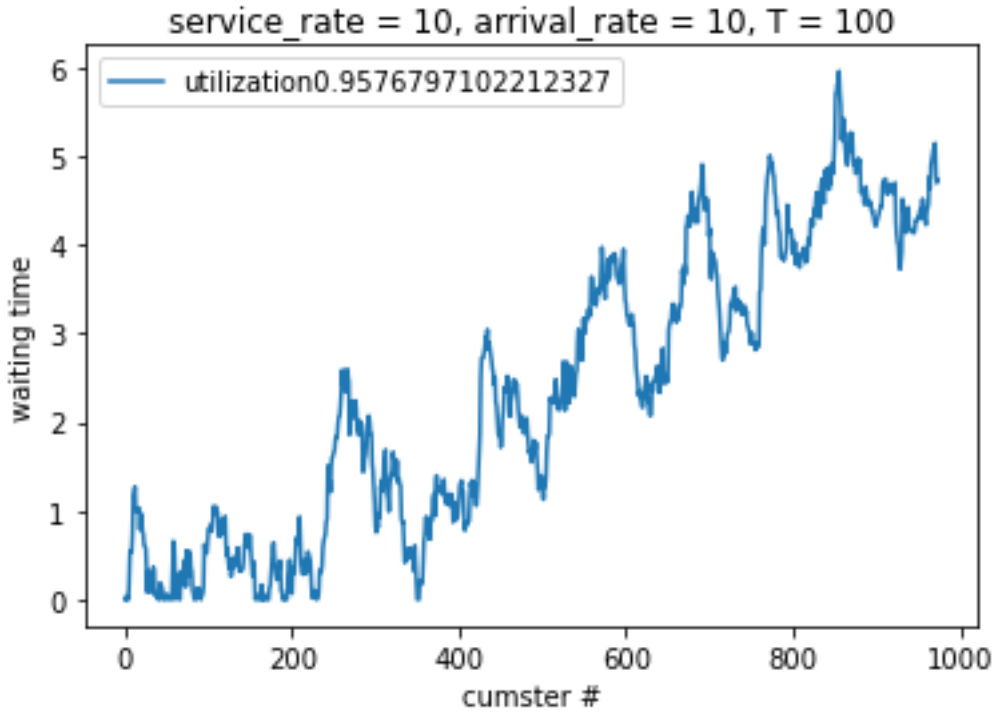

Fig. 7 λ = u

- λ > u (arrival rate > service rate):

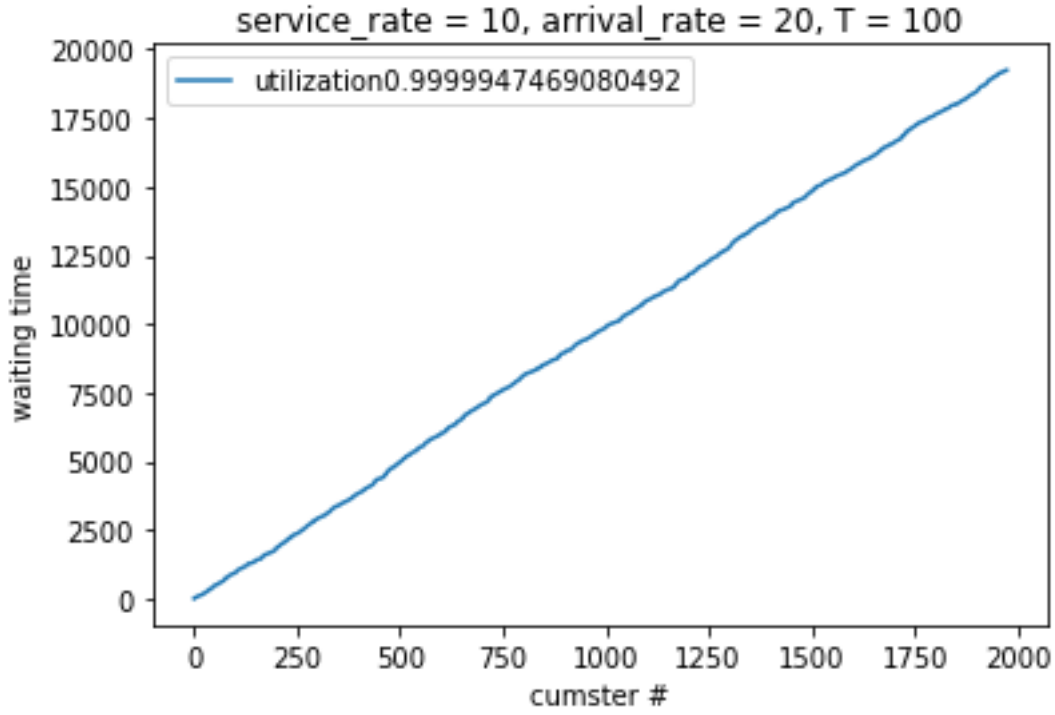

Fig. 8 λ > u

6-1 討論:

從上面的模擬結果可以觀察到幾點:

- 根據對上面的模擬結果，我們能發現在 λ < u 時，Utilization 都非常接近 λ/u，不過在 λ = u 時卻沒有與理論值相符，要到 λ > u 才會趨近於 1，也就是說在超載的情況下 Utilization = 1，always busy。
- 根據上面的模擬結果，可以發現 waiting time 會因為 λ/u 越高，越容易會出現排隊的情形，而 waiting time 也會因為開始排隊，越後面的客人 waiting time 越高。
- 固定時間的情況下，arrival rate 會影響來客人數。
- 議題一:Can utilization go to 1?

  依據上面的模擬結果顯示當 λ > u 時 utilization 會非常趨近 1，也就是超載了。
- 議題二: Why plot the distribution of waiting time, but not the histogram?

  我認為等待時間可能會與前後有關係，因此繪製 the distribution of waiting time 會比較能觀察到前後關聯性。

Homework (It is simple, submit it next week)

- Add the confident interval for your M/M/1 simulation.
- Discuss the different between the simulation round n.

6-2 模擬說明

利用 6-1 的 M/M/1 模擬程式計算 average waiting time 並加上 95%的信賴區間，並且觀察模擬次數對新賴區間的影響。

設定：λ(arrival rate)= 1~20，u(service rate) = 5、10、15，n(模擬次數) = 5、10、100。

6-2 程式碼:

```python
def CI_95(waiting_time):
    '''95信賴區間計算及平均值'''
    waiting_time = np.array(waiting_time)
    mean = np.mean(waiting_time)
    std = np.std(waiting_time)
    r = 1.96 * std/(len(waiting_time)**(1/2))
    return mean, r
```

Fig. 9 信賴區間計算

```python
def draw(num):
    '''製圖'''
    mean_1 = []
    mean_2 = []
    mean_3 = []

    listci_1 = []
    listci_2 = []
    listci_3 = []

    lam_L = np.arange(1, 21, 1)

    for lam in lam_L:
        temp_list1 = []
        temp_list2 = []
        temp_list3 = []
        for i in range(num):
            arrival_time, service_time, waiting_time, departure_time, _i = MM1_model_fixed_time_interval(5, lam, 10)
            arrival_time2, service_time2, waiting_time2, departure_time2, _i2 = MM1_model_fixed_time_interval(10, lam, 10)
            arrival_time3, service_time3, waiting_time3, departure_time3, _i3 = MM1_model_fixed_time_interval(15, lam, 10)

            temp_list1.append(np.sum(waiting_time) / _i)
            temp_list2.append(np.sum(waiting_time2) / _i2)
            temp_list3.append(np.sum(waiting_time3) / _i3)

        # service rate= 5
        mean1, r1 = CI_95(temp_list1)
        mean_1.append(mean1)
        listci_1.append(r1)

        # service rate= 10
        mean2, r2 = CI_95(temp_list2)
        mean_2.append(mean2)
        listci_2.append(r2)

        # service rate= 20
        mean3, r3 = CI_95(temp_list3)
        mean_3.append(mean3)
        listci_3.append(r3)
```

Fig. 10 average waiting time +信賴區間及製圖-1

```
    color = ['r', 'g', 'b']
    plt.plot(lam_L, mean_1, label="service rate = 5", color=color[0])
    plt.fill_between(lam_L, np.subtract(mean_1, listci_1), np.add(mean_1, listci_1), color=color[0], alpha=0.5)

    plt.plot(lam_L, mean_2, label="service rate = 10", color=color[1])
    plt.fill_between(lam_L, np.subtract(mean_2, listci_2), np.add(mean_2, listci_2), color=color[1], alpha=0.5)

    plt.plot(lam_L, mean_3, label="service rate = 15", color=color[2])
    plt.fill_between(lam_L, np.subtract(mean_3, listci_3), np.add(mean_3, listci_3), color=color[2], alpha=0.5)

    plt.xlabel('lambda')
    plt.ylabel('waiting time')
    plt.title('n = '+str(num))
    plt.legend()
    plt.show()

draw(5)
draw(10)
draw(100)
```

Fig. 11 average waiting time +信賴區間及製圖-2

6-2 模擬結果:

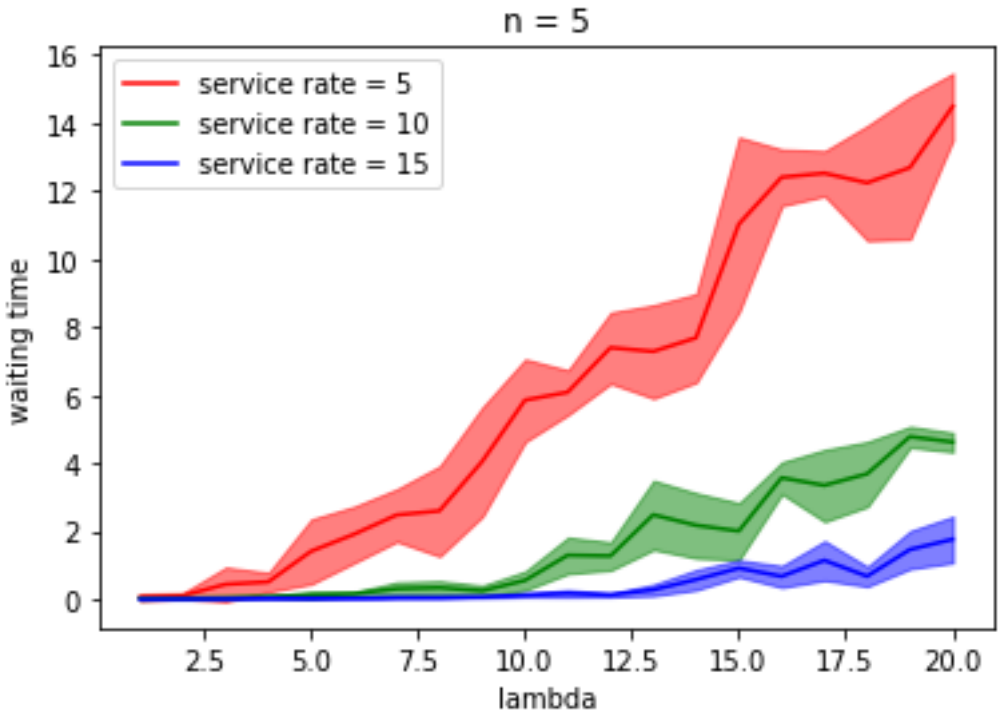

Fig. 12 n = 5 (x: service time y: average waiting time)

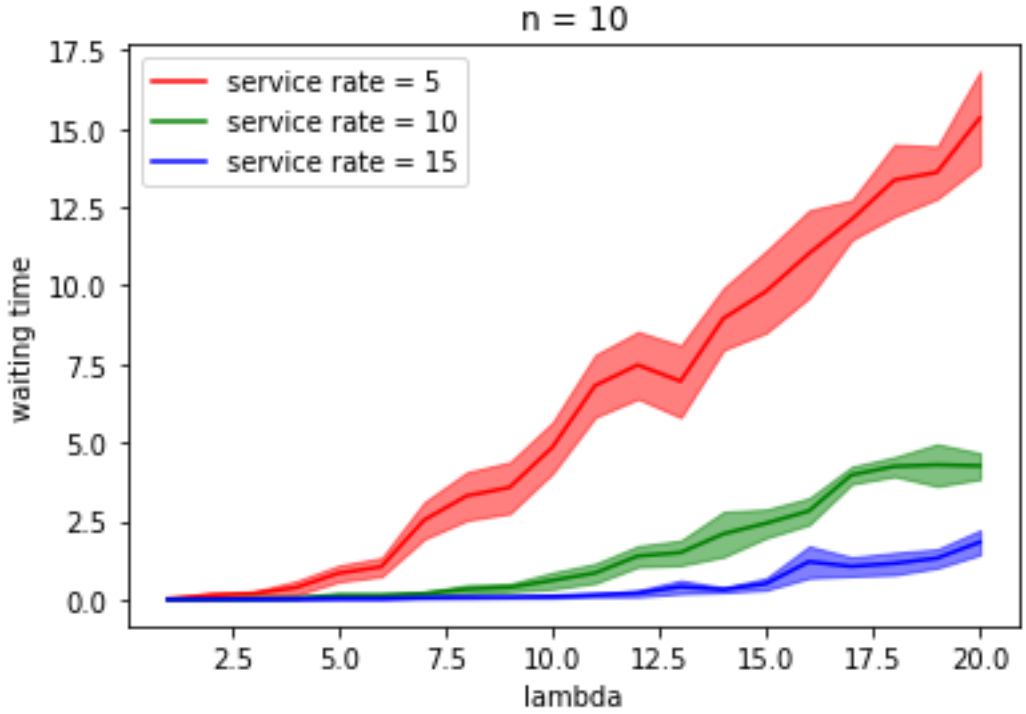

Fig. 13　n = 10 (x: service time y: average waiting time)

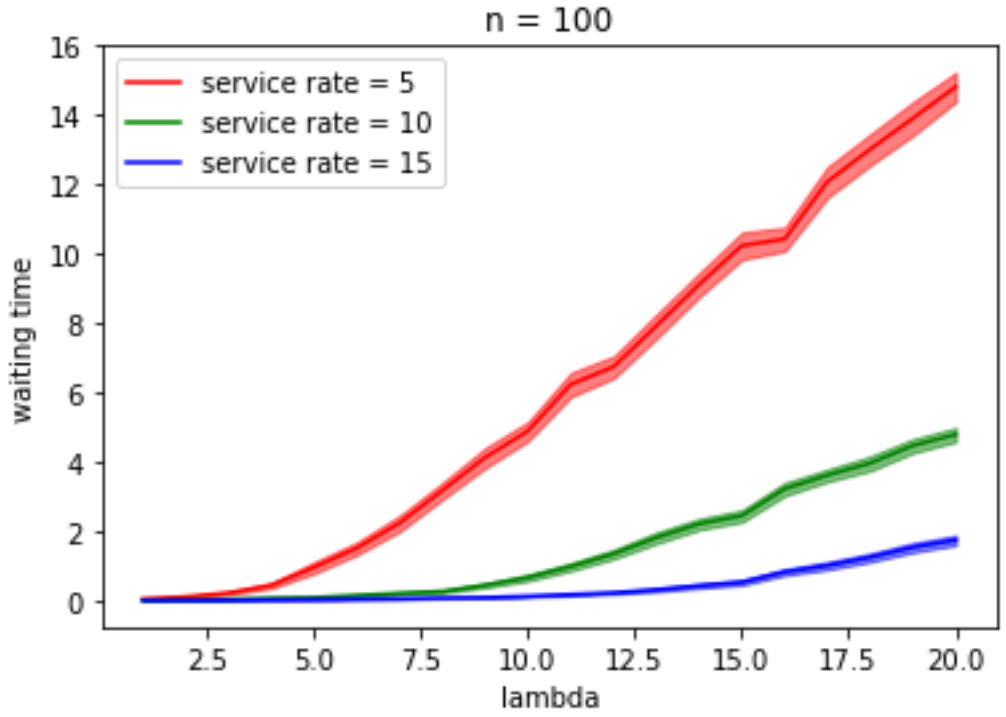

Fig. 14　n = 100 (x: service time y: average waiting time)

6-2 討論:

從上面的模擬結果可以觀察到幾點:

- 固定 service rate 時，arrival rate 越高 average waiting time 越高。

- 固定 arrival rate 時，service rate 越高 average waiting time 越低。
- 增加模擬次數圖形會比較平滑，且信賴區間的範圍也會隨之變小。
- Average waiting time 曲線開始上揚之區域為 λ＝u 的時候(因為會開始排隊，導致後面的客人等越來越久)。