

Queue-Wait-Time for MM1

W. M. Song 桑慧敏

Tsing Hua Univ. 清華大學

2015.10.20

- 1 Goal and Model
- 2 Global Tables
- 3 Flexsim Functions
- 4 Stopping Rule: No. arrivals
- 5 Record Queue-Wait Time
- 6 Output Data
- 7 Homework

- **Goal:** To obtain the queue-wait-time for the i^{th} customer, $W_Q(i)$. $i = 1, 2, \dots, n$
- **Model:** MM1 with arrival rate and service rate stored in Global Tables, named "Parameters".
 - Construct MM1 via Flexsim
 - Construct Global Table (see page 4)
 - Learn New Flexsim Functions (see page 5)
 - Create and Initialize Label (See page 7)
 - Learn stopping rule, no. of arrivals (See page 7)
 - Set label number (See pages 8 - 10)
 - Record W_Q (See Page 12)

Data in Global Tables

Tables

- W_Q_Table
- Total_Arrivals
- Parameters
- Result

The screenshot displays four overlapping windows of Global Tables in Flexsim. Each window has a 'Name' dropdown menu highlighted with a red box:

- Global Table - W_Q_Table**: Name: W_Q_Table, Rows: 5000, Columns: 3.00, Clear on Res: checked. Table content:

	enter Queue	left Queue	wait time in Queue
Row 1	0.00	0.00	0.00
Row 2			
Row 3			
Row 4			
Row 5			
Row 6			
Row 7			
- Global Table - Total_Arrivals**: Name: Total_Arrivals, Rows: 1.00, Columns: 1.00, Clear on Res: unchecked. Table content:

	Col 1
Row 1	5000.00
- Global Table - Parameters**: Name: Parameters, Rows: 2.00, Columns: 4.00, Clear on Res: unchecked. Table content:

	location	scale	shape	stream
arrival parameters	0.00	0.50	1.00	0.00
service parameters	0.00	0.33	1.00	0.00
- Global Table - Result**: Name: Result, Rows: 1.00, Columns: 1.00, Clear on Res: unchecked. Table content:

	Average Wait Time in Q
	0.72

Flexsim Data Types and Functions

Data Type

- **int**: integer (整數)
- **double**: real value (實數)
- **string**: (字串)
- **treenode**: (Flexsim 特有的 Data Type)

Flexsim Functions

- **getoutput**(current)
- **getinput**(current)
- **getlabelnum**(item,"labelname")
- **setlabelnum**(item,"labelname";time())
- **gettabelnum**("tablename"; i-row, j-column)
- **settabelnum**("tablename"; i-row, j-column, enter-queue-time)

Generate Inter-arrival Times

- Weibull with para. location, scale, and shape
- Parameters are stored in Global Tables

The image shows two windows from the Flexsim software. The left window is titled "Source1 Properties" and has the "Source" tab selected. The "Arrival" dropdown is set to "Inter-Arrival Time" and the "FlowItem" is "Box". The "Inter-Arrival" dropdown is set to "weibull(location, scale, shape, stream)". The right window is titled "/Source1 - Inter-Arrival Time" and displays the following code:

```

1 treenode current = ownerobject(c);
2 /**popup:StatisticalDistribution*/
3
4 double location;
5 double scale;
6 double shape;
7 double stream;
8
9 location = gettablenum("Parameters",1,1);
10 scale = gettablenum("Parameters",1,2);
11 shape = gettablenum("Parameters",1,3);
12 stream= gettablenum("Parameters",1,4);
13
14 return
15 /**tag:distribution/**weibull/**/**/ (**/
16 /**tag:par1/**/**/location/**/
17 /**tag:par2/**/**/ scale/**/
18 /**tag:par3/**/**/ shape/**/
19 /**tag:par4/**/**/ stream/**/
20 /**tag:par5/**/**/**/
21 /**/**/**/
22 ;
23
  
```

Stop Rule: Total arrivals

- Source > Trigger > OnCreation
- Add "Create and Initialize" twice: enter-queue-time; left-queue-time
- Reset > Run > Stop, then click "item" to see what happens

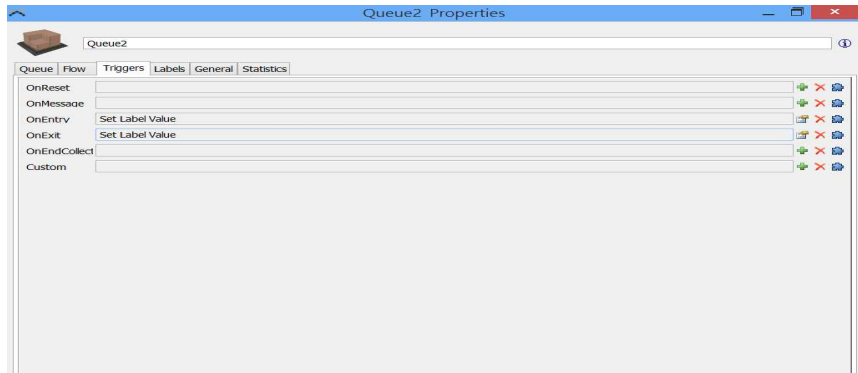
The image shows two windows from the Flexsim software. The left window is the 'Source1 Properties' dialog, with the 'Triggers' tab selected. Under the 'OnCreation' trigger, the 'Create and Initialize Label, Create' option is checked. The right window is the custom code editor for the 'OnCreation' trigger, showing the following code:

```

1 /*Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int rownumber = parval(2); //row number of the schedule/seq
5
6 {
7 treenode involved = /**tag:involved/**/item/**/;
8 string labelname = /**tag:labelname/**/"enter_queue_time"
9 double newvalue = /**tag:newvalue/**/0/**/;
10
11 set(assertlabel(involved,labelname,DATATYPE_NUMBER),newvalue
12)
13 /******* PickOption End *****/
14 /******* PickOption Start *****/
15 {
16 treenode involved = /**tag:involved/**/item/**/;
17 string labelname = /**tag:labelname/**/"left_queue_time"/
18 double newvalue = /**tag:newvalue/**/0/**/;
19 set(assertlabel(involved,labelname,DATATYPE_NUMBER),newvalue
20)
21 /******* PickOption End *****/
22 /**=====written code start=====
23 int n_Arrival;
24 n_Arrival = gettablenum("Total_Arrivals",1,1);
25 if(getoutput(current)==n_Arrival)
26 {
27     stopoutput(current); /******* PickOption Start ***
28 }
29 /**=====written code end=====
  
```

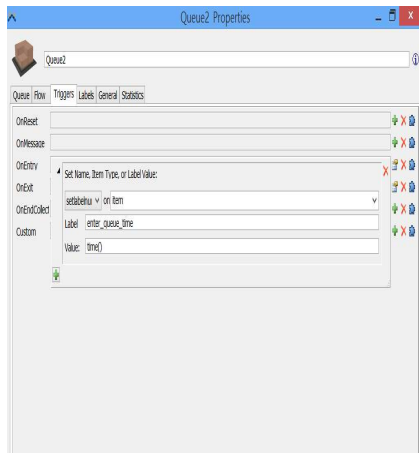
Record time entering and leaving Queue

- Queue > Trigger > OnEntry
- Queue > Trigger > OnExit



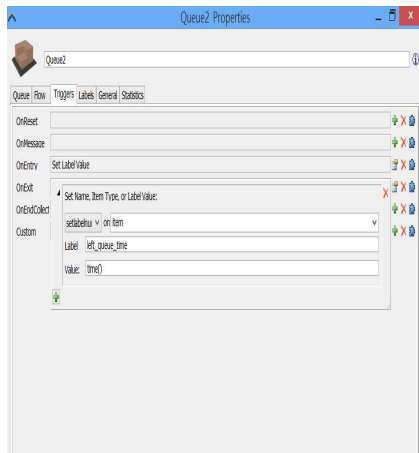
Record Time Entering Queue

- `setlabelnum (item, "enter_queue_time", time())`



Record Time Leaving Queue

- `setlabnum (item, "left_queue_time", time())`



Generate Service Times

- Weibull with parameters location, scale, and shape
- Parameters are stored in Global Tables

The image shows two windows from the Flexsim software. The left window is titled "Processor3 Properties" and has the "Processor" tab selected. The "Processor" tab contains several settings:

- Processor:** Processor3
- Maximum:** 1.00
- Setup Time:** 0
- Process:** weibull(location, scale, shape, stream)
- Pick:** centerobject(current, 1)

 The "OK" button at the bottom right of this dialog is highlighted with a red box.

The right window is titled "/Processor3 - Process Time*" and displays the following code:


```

1 treenode current = ownerobject(c);
2 treenode item = parnode(1);
3
4 double location;
5 double scale;
6 double shape;
7 double stream;
8
9 location = gettablenum("Parameters",2,1);
10 scale = gettablenum("Parameters",2,2);
11 shape = gettablenum("Parameters",2,3);
12 stream = gettablenum("Parameters",2,4);
13
14 return
15 /**tag:distribution***/weibull/**/**/ (**/
16 /**tag:par1***/location/**/
17 /**tag:par2***/scale/**/
18 /**tag:par3***/shape/**/
19 /**tag:par4***/stream/**/
20 /**tag:par5***/);
    
```

 The "OK" button at the bottom right of this code editor is also highlighted with a red box.

Record W_Q

- Sink > Trigger > OnEntry
- $n = \text{getinput}(\text{current})$
- $\text{getlabelnum}(\text{item}, \text{"enter_queue_time"})$;
- $\text{settablenum}(\text{"W_Q_Table"}, n, 1, \text{enter_queue_time})$

The image shows two windows from the Flexsim software. The left window is titled 'Sink4 Properties' and shows the 'OnEntry' trigger set to 'Custom Code'. The right window is titled '/Sink4 - OnEntry*' and displays the following custom code:

```

1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 int n;
7 int i;
8 double wait_time_in_system=0;
9 double average_wait_time;
10 double enter_queue_time;
11 double left_queue_time;
12 //double lenght_in_system;
13 n = getinput(current);
14 enter_queue_time = getlabelnum(item,"enter_queue_time");
15 left_queue_time = getlabelnum(item,"left_queue_time");
16 settablenum("W_Q_Table",n,1,enter_queue_time);
17 settablenum("W_Q_Table",n,2,left_queue_time);
18 settablenum("W_Q_Table",n,3,left_queue_time-enter_queue_time);
19
20 if(n==gettablenum("Total_Arrivals",1,1))
21 {
22     for(i=1;i<gettablenum("Total_Arrivals",1,1);i++)
23     {
24         wait_time_in_system = wait_time_in_system + gettablenum("W_Q_Table",i,3);
25     }
26     average_wait_time = wait_time_in_system/gettablenum("Total_Arrivals",1,1);
27     settablenum("Result",1,1,average_wait_time);
28 }
29

```

"if" and "for" Commands in Flexsim

```
if(n == gettablenum("Total_Arrivals",1,1))
{
for(i=1; i <= gettablenum("Total_Arrivals",1,1); i++)
{
wait_time_in_system = wait_time_in_system +
gettablenum("W_Q_Table",i,3);
}
average_wait_time =
wait_time_in_system/gettablenum("Total_Arrivals",1,1);
settablenum "Result",1,1,average_wait_time);
endreplication(1);
}
```

Replications

Simulation Experiment Control

Scenarios Performance Measures Experiment Run Advanced

Scenario 5 Experiment 6 go to Scenario

Variable	Scenario	Scenario	Scenario	Scenario	Scena
location	/Tools/GlobalTables/parameters> variables/data/1/1	0	0	0	0
scale	/Tools/GlobalTables/parameters> variables/data/1/2	30	30	30	30
Variable 3	variables/data/1/3	1	2	3	4
Variable 4	Table Value	variables/data/2/1	0	0	0
Variable 5	parameters	variables/data/2/2	20	20	20
Variable 6	Row 1	variables/data/2/3	1.00	2.00	3.00
	Column			4.00	5

1
2
3

Replications

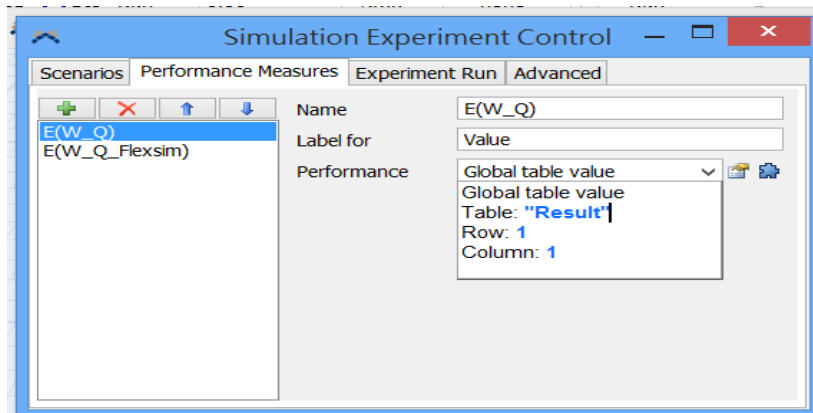
Simulation Experiment Control

Scenarios Performance Measures Experiment Run Advanced

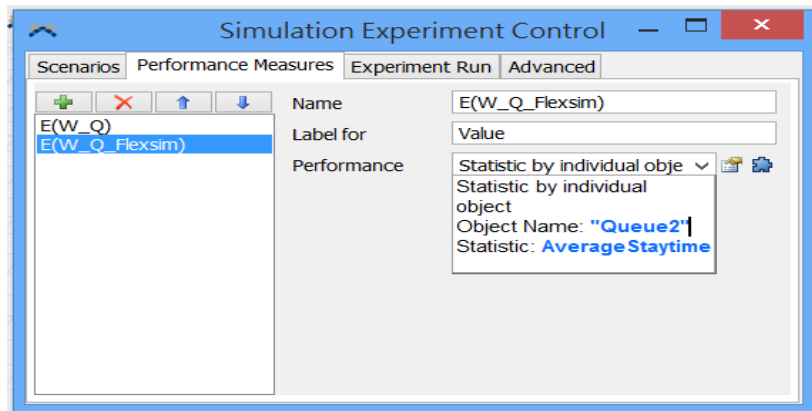
Scenario 5 Experiment 6 go to Scenario

Variable	Scenario	Scenario	Scenario	Scenario	Scenario
location	/Tools/GlobalTables/parameters>variables/data/1/1	0	0	0	0
scale	/Tools/GlobalTables/parameters>variables/data/1/0	30	30	30	30
Variable 3	/Tools/GlobalTables/parameters>variables/data/1/3	1	2	3	4
Variable 4	Table Value	variables/data/2/1	0	0	0
Variable 5	parameters	variables/data/2/2	20	20	20
Variable 6	parameters	variables/data/2/3	1.00	2.00	3.00
Row	1				
Column					
	1				
	2				
	3				

Replications



Replications



Replications

The screenshot shows the 'Simulation Experiment Control' window with the 'Advanced' tab selected. The 'Experiment Run' tab is also visible. The settings are as follows:

- Run to: 360000.
- Warmup: 0.00
- Replications per: 30.00
- Buttons: **Reset Experiment**, **View Results**, **Export/Merge Results** (dropdown), and **Export results after each replication** (checkbox).
- Options:
 - Save dashboard data for each replication
 - Save state after each replication
 - Export results after each replication
- Experiment Status** section showing progress bars for Scenario 1 through Scenario 5, all of which are fully completed (green bars).

Output Analysis

E(W_Q)

Summary					
	Mean (90% Confidence)		Sample Standard Deviation	Min	Max
Scenario 1	38.78	< 40.79 < 44.79	12.91	24.03	88.27
Scenario 2	7.07	< 7.53 < 7.98	1.47	5.64	12.04
Scenario 3	2.65	< 2.78 < 2.87	0.36	2.04	3.59
Scenario 4	1.17	< 1.22 < 1.27	0.17	0.82	1.64
Scenario 5	0.58	< 0.6 < 0.63	0.08	0.41	0.74

E(W_Q_Flexsim)

Summary					
	Mean (90% Confidence)		Sample Standard Deviation	Min	Max
Scenario 1	38.78	< 40.79 < 44.79	12.91	24.03	88.27
Scenario 2	7.07	< 7.53 < 7.98	1.47	5.64	12.04
Scenario 3	2.65	< 2.78 < 2.87	0.36	2.04	3.59
Scenario 4	1.17	< 1.22 < 1.27	0.17	0.82	1.64
Scenario 5	0.58	< 0.6 < 0.63	0.08	0.41	0.74

Dependence and Correlation

C2: fx "= CORREL(A2:A1000,B2:B1000)"

	A	B	C
1	wait time in Queue(i)	wait time in Queue(i+1)	
2	0	0	0.773595224
3	0	0	
4	0	11.14	
5	11.14	0	

A

995	0	19.29	
996	19.29	0	
997	0	14.41	
998	14.41	0	
999	0	0	
1000	0	0	

Homework

- Generate time in system (rather than queue-wait-time) of M/M/1 for each customer, say $W(i), i = 1, 2, \dots$
- Check whether $W(i), i = 1, 2, \dots$ are independent or dependent
- Check whether $W(i), i = 1, 2, \dots$ follow the same distributions