McDonald’s Customer’s Experience

Project for DAT530

(Discrete Simulation & Performance Analysis)

By: Bilal Amjad

Department of Computer Science & Technology

University of Stavanger

Table of Contents

ABSTRACT:

There is a wide range of fast food restaurants out there in the market nowadays, but I have chosen McDonald's for my project. In this project, I have simulated all the processes that a McDonald's customer goes through in relation to buying food for himself. For simulation and evaluation of these processes, I have made use of MATLAB and GPenSIM. As, I have three types of customers involved that's why I have used the technique of colored petri net for the categorization of the tokens. Whereas, testing and analysis results give us a meaningful insight of sales, production time and various likewise aspects related to the restaurant.

1.INTRODUCTION

McDonald's is a multinational food chain which offers breakfast and food services all day long. Breakfast, lunch and dinner is a busy time for any restaurant universally. My project is about customer's process at McDonald's which includes selection of food, payment and eating i.e. either dine-in or take away. We have three major type of customers namely:

- One who buy and eat there called Dine Inn customers
- Second, who buy and takes away called take away customers
- Third, who uses drive-thru service called drive thru customers

I have also taken two more types of customers into consideration:

- Who leaves the order queue from within the restaurant (Dine Inn / Take away)
- Who leaves the order queue from Drive Thru service

1.1 Problem Definition:

I have used GPenSIM and Petri NET to simulate the process of all three above stated types of customers. I analyzed peak hours and off-peak hours of a restaurant. So that the customers can avoid the rush hours to get their orders in a short time. Also, I got an insight of how many customers (Dine Inn / Take away and Drive Thru) are leaving the order queues due to various reasons and how many orders are waiting or pending to go into production at any given time. Moreover, I could differentiate between number of customers on weekdays and weekend.

These kind of analysis and statistics are very crucial to the management because that's how they can gauge their performance and eliminate their deficiencies, if any.

1.2 McDonald's Customer's Process:

1.2.1 Dine Inn Customer: Customer enters the restaurant. Makes up his mind about what he wants to eat and go straight to the automatic order machine also known as kiosk machine. He waits in the queue, provided if there is a person already ordering his food via machine. Then he chooses his order, pay his bill and gets the receipt generated by machine. He then waits until his order is prepared and displayed as ready for delivery on the screen, fitted right above delivery counter. He collects his order after it is ready and go straight to the dining area of the restaurant to enjoy his meal.

1.2.2 Take Away Customer: The process of take away customer is same as Dine Inn customer except the fact that after taking delivery of his order, he does not eat there. Rather he takes it and leave the restaurant.

1.2.3 Drive Thru Customer: Drive Thru service is for them who don't want to get off their vehicle and they can order and get their food delivery while sitting in their cars. They have a separate order queue outside the restaurant where they can order and on the next station/point they can pay their bill and take delivery of their food. After that they can just leave or enjoy food in the parking.

1.2.4 Leaving Customers: Customers who leave the order queue. It includes customers leaving from inside the restaurant (Take Away & Dine In customers) and from outside the restaurant i.e., from Drive Thru order queue.

1.3 Adaptations:

I have adapted the token generation module from a past project named as "The Drop Zone by Eric Freeman, Ole Tobiesen and Gabriel Nketah". Also, I took guidance on how to design my model from another past project named as "Student Canteen Process Simulation by Wei Liao & Long Cui" because to some extent it was related to the customers who buy food and eat there or takeaway.

2. Method and Design:

2.1 Overall Design:

My project consists of five major modules namely: Token generation module, Dine Inn customer module, Takeaway customer module, Drive Thru customer module and Leaving module for leaving customers. All these modules are knitted well together to form an overall design of the working model.

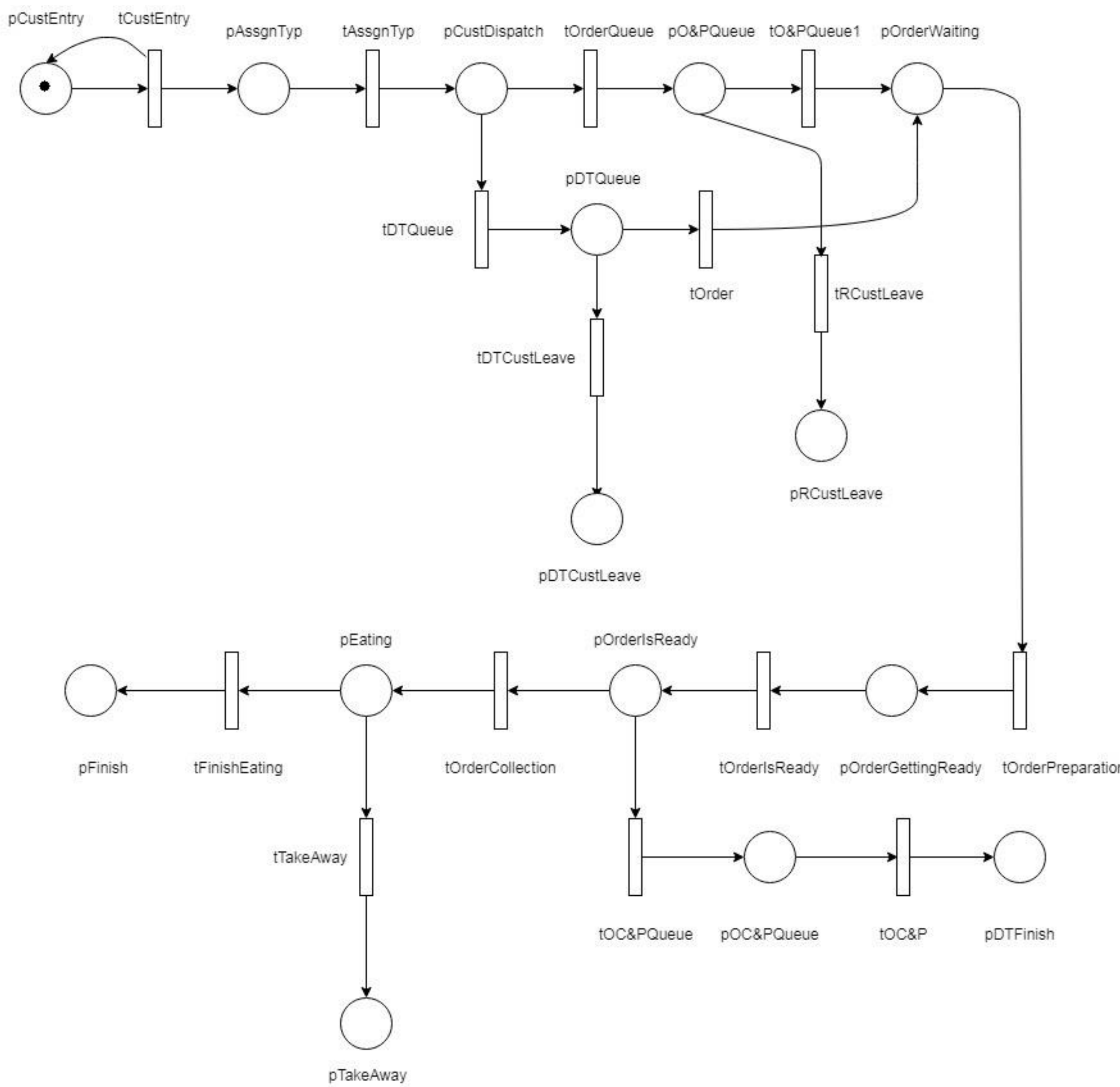

Figure 1: Full Scale Model

2.2 Modular Approach:

Above five stated modules of my model are described as follows:

2.2.1 Token generation module: It generate tokens as customers at a random time. After that it assigns them a color randomly as a customer type for the purpose of classification.

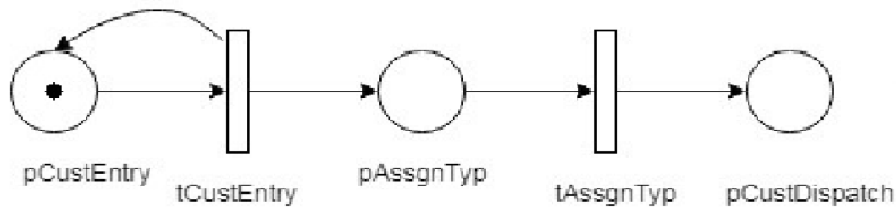

Figure 2: Token Generation Module

2.2.2 Dine Inn customer module: It depicts all the steps involved with a dine inn customer, from ordering of the food to the delivery and finally eating in the dining area.

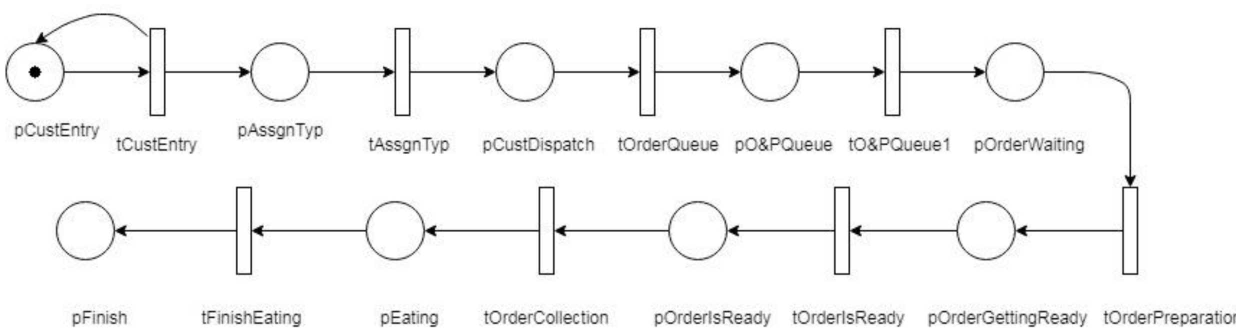

Figure 3: Dine Inn customer module

2.2.3 Takeaway customer module: It has all the same steps involved as a dine inn customer but only until delivery of the food. After getting the food this customer just leaves the place.

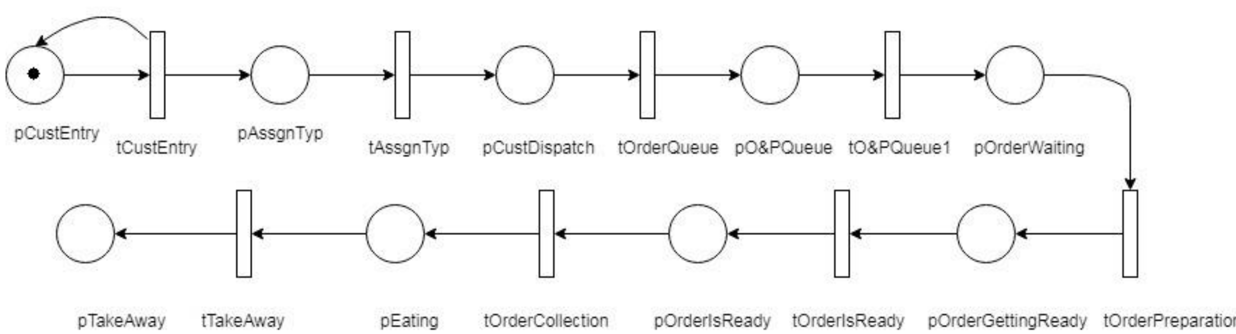

Figure 4: Take Away customer module

2.2.4 Drive Thru customer module: This type of customer has different physical place for ordering the food, which is shown and simulated in my model. But production area for orders is same for all types of customers. Also, for taking food delivery and payment, these customers have another physical place or station and after this they leave.

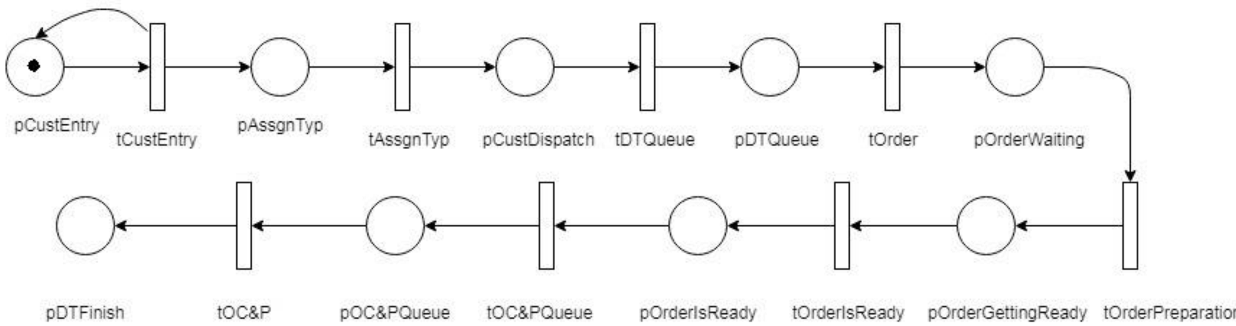

Figure 5: Drive Thru customer module

2.2.5 Leaving module: This module represents customers who leave the order queue i.e., from within the restaurant (dine in or take away customers) or from the drive thru order queue situated outside the restaurant.

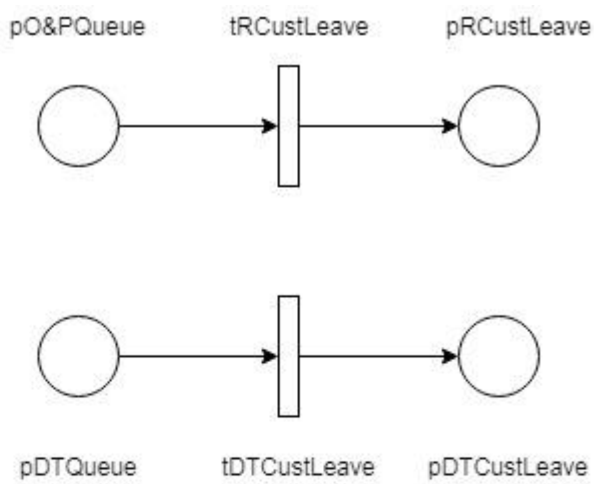

Figure 6: Leaving module

2.3 Technique:

I have used colored petri technique in my project for simulation. After the generation of customers through token generator, customers are categorized based on token colors using color petri net technique. It enabled me to do analysis based on token color and gave me idea that which type of customer is visiting restaurant more often on daily basis as compared to other types.

3. Implementation:

Since, I was unable to acquire required and relatable data for my project, from any source that's why I am running the simulations for time that the restaurant is operational on a normal weekday i.e. sixteen hours which make up to 57600 seconds. Also, I have assigned probabilities to the five types of customers.

```
%16 hours working shift of the restaurant is represented by 576000 seconds
global_info.STOP_AT = 57600;
%represents 5 types of customers in the form of colors
%'RCustLeave' includes DineInn & TakeAway Leaving customers while
%'DTCustLeave' includes DriveThru Leaving customers
global_info.CustomerType = {'DineInn', 'TakeAway', 'DriveThru','RCustLeave','DTCustLeave'};
%It is the probability of 5 types of customers
global_info.CustomerProbability = [0.2 0.2 0.2 0.2 0.2 ];
```

Code snippet 1

3.1 Full Scale Model:

My model consists of five modules which includes the following procedures:

1. arrival of customer into the restaurant & assigning one of the five colors randomly
2. food buying process for a Dine Inn customer
3. food buying process for a Take Away customer
4. food buying process for a Drive Thru customer
5. leaving steps for Dine Inn/ Take Away and Drive Thru customer from the restaurant

Above stated modules are described in detail along with their respective code in the following.

3.1.1 Coding of modules:

3.1.1.1 Customer Generator Module:

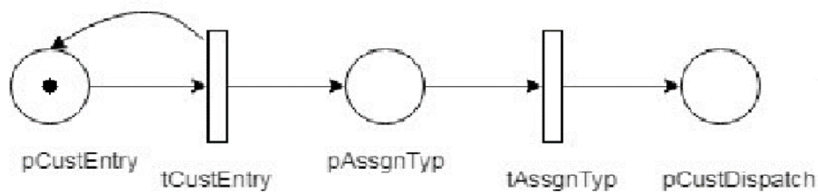

Figure 7: Customer Generator Module

Customer generator module consists of three places (*pCustEntry, pAssgnTyp, pCustDispatch*) and two transitions (*tCustEntry, tAssgnTyp*). Tokens are generated at random times and later tokens are assigned colors randomly based on given customer probabilities. Whereas token color represents the type of a customer.

```
% time and counter values are used for the token generation
time = 0;
counter = 0;
global_info.TokenGeneration_Times = zeros(1, 20000); %An array having token generation times
% For the generation of customers untill the specified time
while lt(time , 57600)
    counter = counter + 1;
    time = time + normrnd(410, 200);
    global_info.TokenGeneration_Times(counter) = time;
end
```

Code snippet 2: For token Generation

Transition *tCustEntry* takes a token from *pCustEntry* and generates customers at random times. It also put back a token to *pCustEntry* for the continuation of the simulation.

```
%this transition is for creating a new token in the form of a customer
if (strcmpi(transition.name, 'tCustEntry'))
    % Don't let the transition fire if the array is empty
    if isempty(global_info.TokenGeneration_Times)
        fire = 0;
        return;
    end;
    cTime = current_time(); %to get the current time
    genrationTime = global_info.TokenGeneration_Times(1);
    % Don't let the transition fire if the remaining generation times in the array are equal to 0
    if eq(genrationTime , 0)
        fire = 0;
        global_info.TokenGeneration_Times = [];
        return;
    end;
    % Don't let the transition fire if current time is lower than the generation time
    if lt(cTime , genrationTime)
        fire = 0;
        return;
    end;
    % It is time to fire
    if ge(length(global_info.TokenGeneration_Times), 2)
        global_info.TokenGeneration_Times = global_info.TokenGeneration_Times(2:end);
    else
        global_info.TokenGeneration_Times = [];
    end;
    fire = 1;
```

Code Snippet 3: For Pre-Processor for transition tCustEntry

Transition *tAssgnTyp* takes the token from *pAssgnTyp* and assigns random color to the token. The color assignment is based on customer probabilities.

```
    %this transition is for assigning a random color to the customer/token
elseif(strcmpi(transition.name, 'tAssgnTyp'))

 %Function for getting cumulative distribution of customers
    cumDist = [0 cumsum(global_info.CustomerProbability)];
    %for the random selection of customers in relation to the given...
    %customer probability
    select = rand();
    greater = find(cumDist >= select==1,1);
    transition.new_color = global_info.CustomerType{greater-1};
    transition.override = true;
    fire=1;
```

Code Snippet 4: For Pre-Processor of transition tAssgnTyp

3.1.1.2 Dine inn Customer Module:

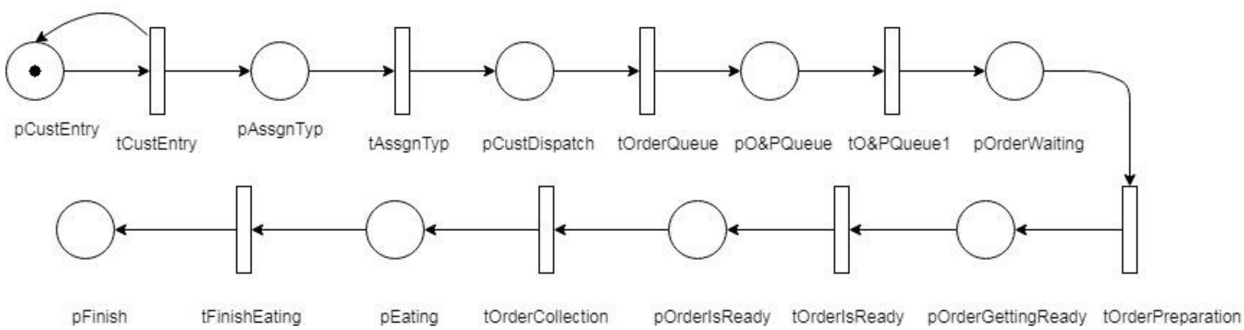

Figure 8: For Dine Inn Customer Module

Having the token colored as *DineInn* in the place *pCustDispatch*, customer goes to the place *pO&PQueue* which is the place for order & pay queue through transition *tOrderQueue*. After that he order and pay for his food through transition *tO&Pqueue1*. His order is now waiting at place *pOrderWaiting* to go into production. Production includes two transitions namely: *tOrderPreparation* and *tOrderIsReady*. Customer collects his order after transition *tOrderCollection* and goes to an eating place *pEating*. Later, he finishes eating by transition *tFinishEating*

The following pre-processor code is used to filter out other two types of customers i.e., *DriveThru* and *DTCustLeave* coming from place *pCustDispatch*,

```
    %this transition is for selecting a token if it's color is one of these i.e., 'DineInn', 'TakeAway','RCustLeave'
elseif(strcmpi(transition.name, 'tOrderQueue'))
    tokID1 = tokenAnyColor('pCustDispatch',1, {'DineInn', 'TakeAway','RCustLeave'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 5: For Pre-Processor of transition tOrderQueue

The following pre-processor code is used to filter out *RcustLeave* type of customer coming from place *pO&Pqueue*

```
    %this transition is for selecting a token if it's color is one of these
    %i.e., 'DineInn', 'TakeAway'
elseif(strcmpi(transition.name, 'tO&PQueue1'))
   tokID1 = tokenAnyColor('pO&PQueue',1, {'DineInn', 'TakeAway'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 6: For Pre-Processor of transition tO&PQueue1

The following pre-processor code is used to select only two types of customers i.e. *DineInn* and *TakeAway* coming from place *pOrderIsReady*

```
    %this transition excludes tokens with color 'DriveThru' and selects
    %tokens only with colors 'DineInn', 'TakeAway'
elseif(strcmpi(transition.name, 'tOrderCollection'))
     tokID1 = tokenWOEXColor('pOrderIsReady',1, {'DriveThru'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 7: For Pre-Processor of transition tOrderCollection

The following pre-processor code is used to select only one type of customers i.e. *DineInn*, coming from place *pEating*

```
    %this transition selects token with exact color i.e., 'DineInn'
elseif(strcmpi(transition.name, 'tFinishEating'))
    tokID1 = tokenEXColor('pEating',1, {'DineInn'});
     transition.selected_tokens = tokID1;
     fire = tokID1;
```

Code Snippet 8: For Pre-Processor of transition tFinishEating

3.1.1.3 Take Away Customer Module:

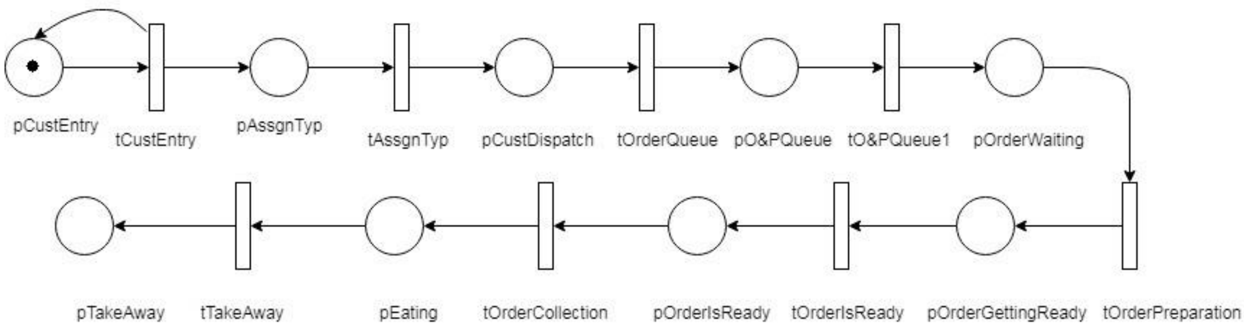

Figure 9: Take Away Customer Module

Having the token colored as *TakeAway* in the place *pCustDispatch*, customer goes to the place *pO&Pqueue* which is the place for order & pay queue through transition *tOrderQueue*. After that he order and pay for his food through transition *tO&Pqueue1*. His order is now waiting at place *pOrderWaiting* to go into production. Production includes two transitions namely: *tOrderPreparation* and *tOrderIsReady*. Customer collects his order after transition *tOrderCollection* and leaves the restaurant via going through eating place using transition *tTakeAway* because he does not eat there.

The following pre-processor code is used to filter out other two types of customers i.e., DriveThru and DTCustLeave coming from place *pCustDispatch*,

```
    %this transition is for selecting a token if it's color is one of these i.e., 'DineInn', 'TakeAway','RCustLeave'
elseif(strcmpi(transition.name, 'tOrderQueue'))
    tokID1 = tokenAnyColor('pCustDispatch',1, {'DineInn', 'TakeAway','RCustLeave'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 9: For Pre-Processor of transition tOrderQueue

The following pre-processor code is used to filter out RcustLeave type of customer coming from place *pO&Pqueue*

```
    %this transition is for selecting a token if it's color is one of these
    %i.e., 'DineInn', 'TakeAway'
elseif(strcmpi(transition.name, 'tO&PQueue1'))
   tokID1 = tokenAnyColor('pO&PQueue',1, {'DineInn', 'TakeAway'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 10: For Pre-Processor of transition tO&PQueue1

The following pre-processor code is used to select only two types of customers i.e. 'DineInn' and 'TakeAway' coming from place 'pOrderIsReady'

```
    %this transition excludes tokens with color 'DriveThru' and selects
    %tokens only with colors 'DineInn', 'TakeAway'
elseif(strcmpi(transition.name, 'tOrderCollection'))
     tokID1 = tokenWOEXColor('pOrderIsReady',1, {'DriveThru'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 11: For Pre-Processor of transition of tOrderCollection

The following pre-processor code is used to select only one type of customers i.e. TakeAway coming from place *pEating*

```
    %this transition selects token with exact color i.e., 'TakeAway'
elseif(strcmpi(transition.name, 'tTakeAway'))
     tokID1 = tokenEXColor('pEating',1, {'TakeAway'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 12: For Pre-Processor of transition of tTakeAway

3.1.1.4 Drive Thru Customer Module:

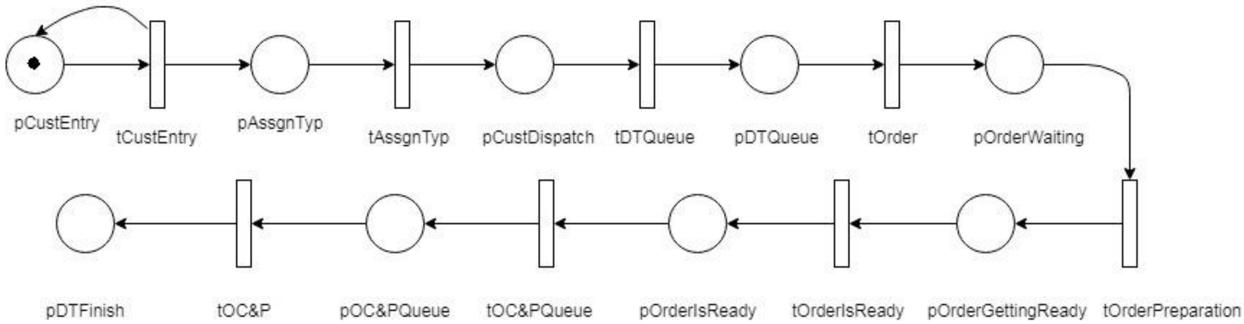

Figure 10: For Drive Thru Customer Module

Having the token colored as DriveThru in the place *pCustDispatch*, customer goes to the place *pDTQueue* which is the place for drive thru order queue through transition *tDTQueue*. After that he orders his food through transition *tOrder*. His order is now waiting at place *pOrderWaiting* to go into production. Production includes two transitions namely: *tOrderPreparation* and *tOrderIsReady*. Customer goes to the place *pOC&Pqueue* which is for order collection and payment queue through transition *tOC&Pqueue*. At last, he collects and pay for his order after transition *tOC&P* and leaves the premises.

The following pre-processor code is used to select only two types of customers i.e., DriveThru and DTCustLeave coming from place pCustDispatch,

```
  %this transition is for selecting a token if it's color is one of these i.e., 'DriveThru','DTCustLeave'
elseif(strcmpi(transition.name, 'tDTQueue'))
    tokID1 = tokenAnyColor('pCustDispatch',1, {'DriveThru','DTCustLeave'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 13: For Pre-Processor of transition of tDTQueue

The following pre-processor code is used to select only one type of customers i.e., DriveThru coming from place *pOrderIsReady*

```
    %this transition selects token with exact color i.e., 'DriveThru'
elseif(strcmpi(transition.name, 'tOC&PQueue'))
     tokID1 = tokenEXColor('pOrderIsReady',1, {'DriveThru'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 14: For Pre-Processor of transition of tOC&PQueue

3.1.1.5 Leaving Customers Module:

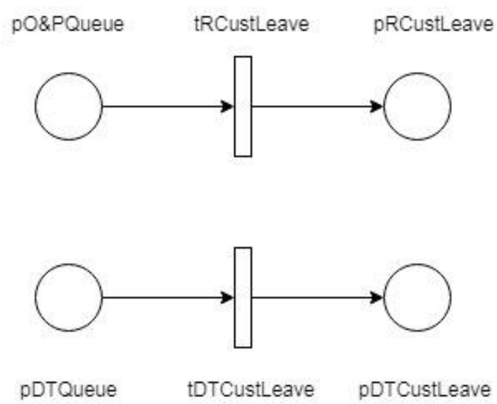

Figure 11: For Leaving Customers Module

This module has only two transitions namely: *tRCustLeave* and *tDTCustLeave*.

RcustLeave represents Dine in and Take Away customers who leaves the order queue.

The following pre-processor code is used to select only one type of customers i.e., RcustLeave coming from place *pO&Pqueue*

```
    %this transition is for selecting a token if it's color is exactly...
    %equal to 'RCustLeave'
    elseif(strcmpi(transition.name, 'tRCustLeave'))
   tokID1 = tokenEXColor('pO&PQueue',1, {'RCustLeave'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
```

Code Snippet 15: For Pre-Processor of transition of tRCustLeave

DTCustLeave represents Drive Thru customers who leaves the Drive Thru order queue. The following pre-processor code is used to select only one type of customers i.e., DTCustLeave coming from place *pDTQueue*

```
%this transition is for selecting a token if it's color is exactly...
%equal to 'DTCustLeave'
elseif(strcmpi(transition.name, 'tDTCustLeave'))
tokID1 = tokenEXColor('pDTQueue',1, {'DTCustLeave'});
transition.selected_tokens = tokID1;
fire = tokID1;
```

Code Snippet 16: For Pre-Processor of transition of tDTCustLeave

3.1.2 Inter- Modular Interaction:

Following is the full-scale model which has all the five modules connected for the successful operation of the restaurant.

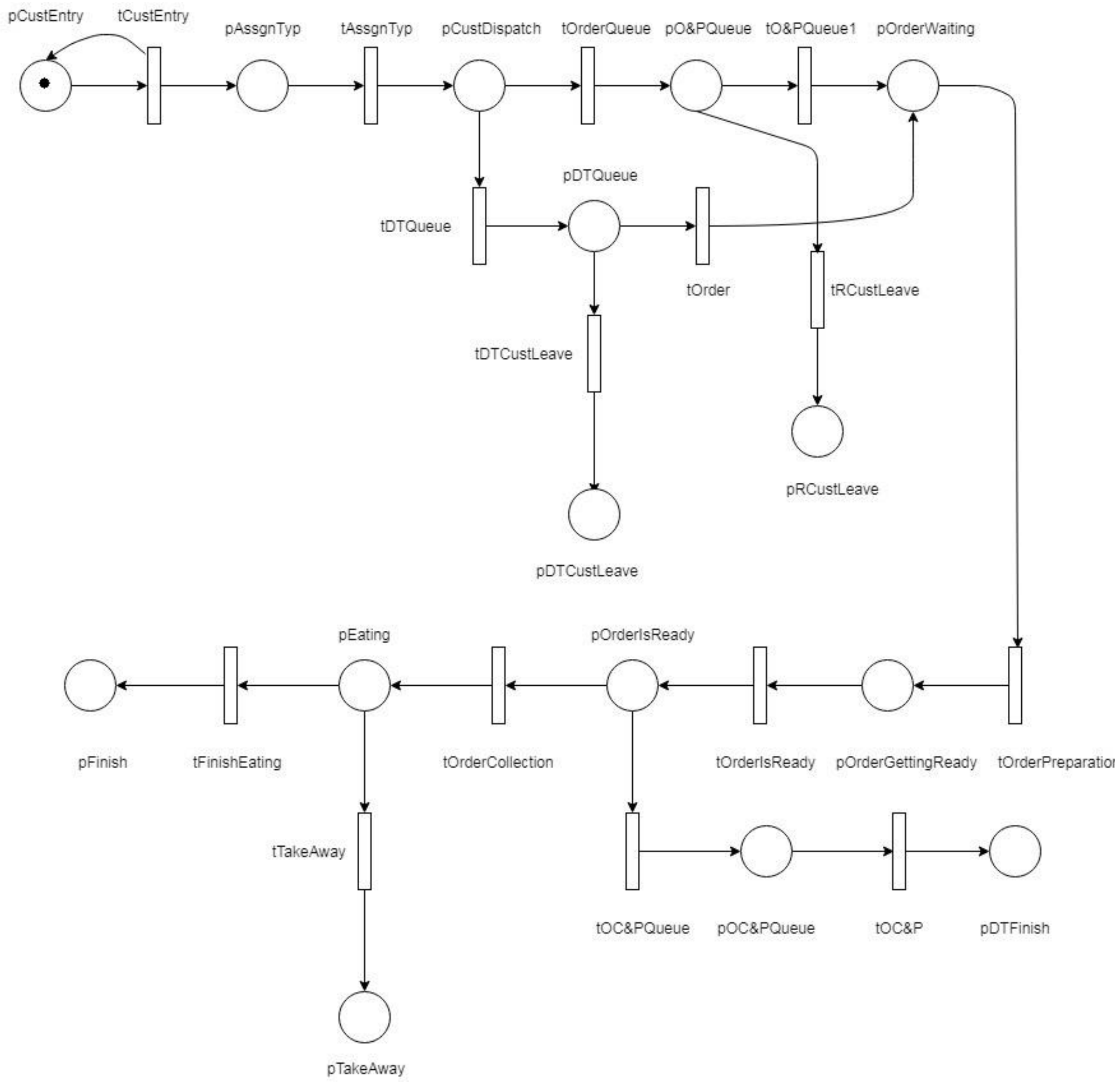

Figure 12: For Inter-Modular Interaction

4.Testing, Analysis & Results:

4.1 Simple User Manual:

1. First install MATLAB and GPenSIM.
2. Unzip the file named as “McDonald’sCustomer’sExperience.rar”
3. This file will have a folder named as “McDonald_Project”.
4. Open this folder in order to run simuation.
5. Open the main simulation file named as “McDonalds.m”.
6. You can make changes as per requirement to the simulation stoppage time and to the customer’s probabilities as shown in Code Snippet 17 below.
7. After making changes, run “McDonalds.m”.

```
global global_info
%16 hours working shift of the restaurant is represented by 576000 seconds
global_info.STOP_AT = 57600;
%represents 5 types of customers in the form of colors
%'RCustLeave' includes DineInn & TakeAway Leaving customers while
%'DTCustLeave' includes DriveThru Leaving customers
global_info.CustomerType = {'DineInn', 'TakeAway', 'DriveThru','RCustLeave','DTCustLeave'};
%It is the probability of 5 types of customers
global_info.CustomerProbability = [0.2 0.2 0.2 0.2 0.2 ];
```

Code Snippet 17: For making parameter changes

4.2 Sample Run: (TU is taken as Seconds in this section)
4.2.1 Plot run for: 57600 TU/16h
Timing: Mon-Fri 7AM -11 PM
Customer Count:189
Simulation Last State: "Current State: pCustEntry + 28pDTCustLeave + 44pDTFinish + 31pFinish + 47pRCustLeave + 38pTakeAway"

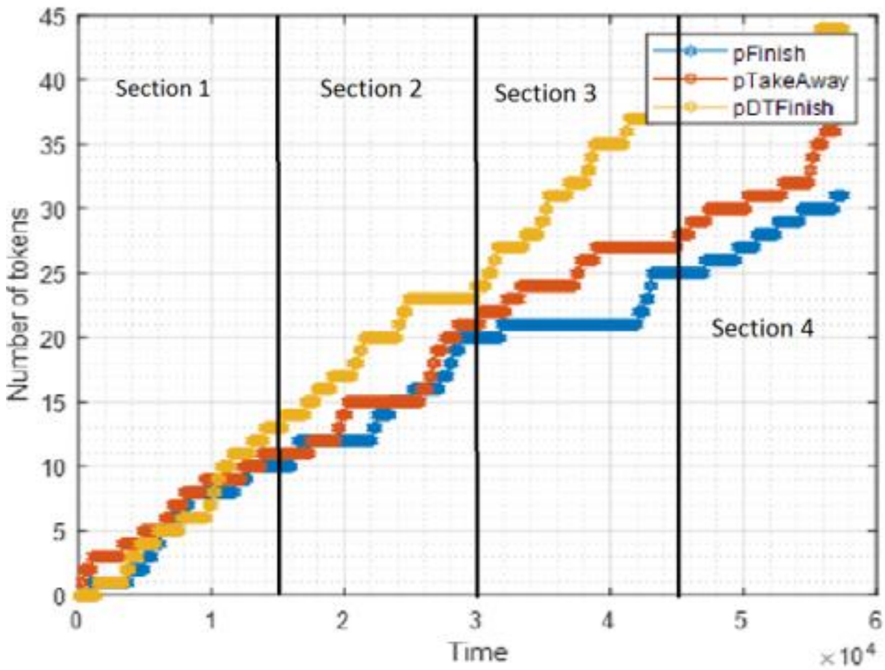

Figure 13: Sample run for 16 hours (Dine In, Take Away & Drive Thru Customers)

This plot represents the single day duration of the restaurant that is sixteen hours on weekdays like Monday to Friday. It has been generated by running simulation for 57600 Tu which takes almost 12 minutes to complete. It's X-axis represents Time which is the working duration of the day and Y-axis shows the number of tokens or customers which were entertained during the whole working duration.

Here I am doing analysis of three types of customers namely: Dine Inn, Takeaway and Drive Thru.

Blue line represents Place *pFinish* which shows Dine Inn customers, red line represents Place *pTakeAway* which shows TakeAway customers while yellow line represents Place *pDTFinish* which shows Drive Thru customers. I have divided the whole daytime in four sections namely: Morning, Afternoon, Evening and Night which are represented by Section 1, Section 2, Section 3 and Section 4 respectively.

Section 1: In the beginning Take Away customers are more in numbers as compared to other customer types. But at the end Drive Thru customers take the lead. Most of the people like the Drive Thru service in this section.

Section 2: Drive Thru customers are greater in number as compared to other customer types from start to end.

Section 3: Drive Thru customers are greater in number as compared to other customer types from start to end. While number of Dine In customers remain constant for some time but elevate at the end of the section.

Section 4: here also Drive thru customers are more than other customer types, take away being the second and dine inn customers are lowest in numbers.

This plot suggests that the peak time of customers for restaurant is the section 4 representing dinner time while section 1 reflects off peak time, having least number of customers.

4.2.2 Plot run for: 54000 TU/15 h

Timing: Sat 8 AM -11 PM

Customer Count: 174

Simulation Last State: "Current State: pCustEntry + 34pDTCustLeave + 38pDTFinish + 40pFinish + 29pRCustLeave + 32pTakeAway"

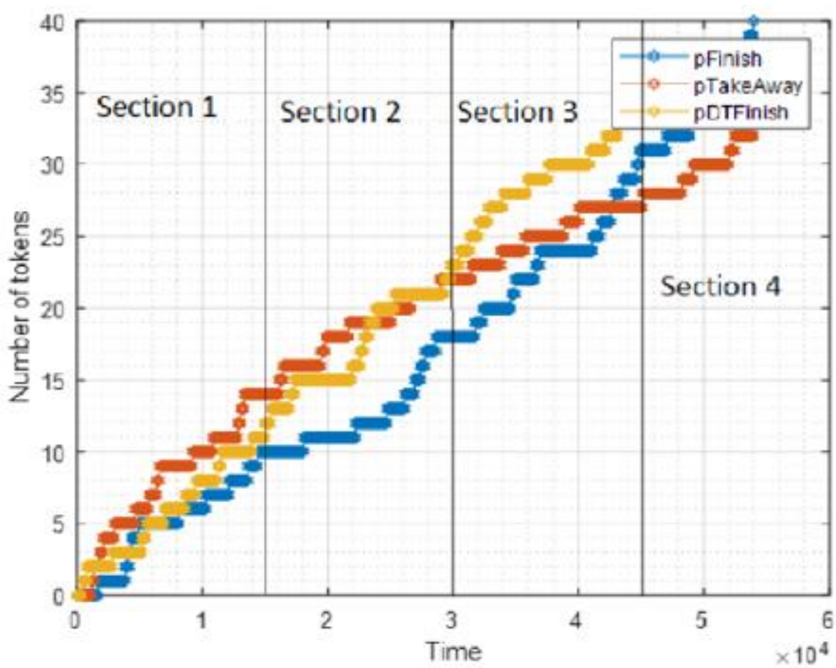

Figure 14: Sample run for 15 hours (Dine In, Take Away & Drive Thru Customers)

This plot represents the single day duration of the restaurant that is fifteen hours on Saturday. It has been generated by running simulation for 54000 Tu. It's X-axis represents Time which is the working duration of the day and Y-axis shows the number of tokens or customers which were entertained during the whole working duration.
Here I am doing analysis of three types of customers namely: Dine Inn, Takeaway and Drive Thru.

Blue line represents Place *pFinish* which shows Dine Inn customers, red line represents Place *pTakeAway* which shows TakeAway customers while yellow line represents Place *pDTFinish* which shows Drive Thru customers. I have divided the whole daytime in four sections namely: Morning, Afternoon, Evening and Night which are represented by Section 1, Section 2, Section 3 and Section 4 respectively.

Section 1: In the beginning there were some Drive Thru customers but after some time number of take away customers took off, being the highest and dine inn customers at the lowest in numbers at the end.

Section 2: The dine inn customers are lowest but shows some consistency over the period of time. At the end both take away and drive thru customers are almost same in numbers.

Section 3: Drive thru customers take the lead in this section. Drive thru customers were at the lowest in the start but gradually increased and becomes second highest in numbers.

Section 4: As this section represents the dinner section so, Dine inn customers are in greater number than other two types.

This plot suggests that the first two sections shows the off-peak time while rest of the two represents the peak time of the day.

4.2.3 Plot run for: 50400 TU / 14 h
Timing: Sun 9 AM -11 PM
Customer Count: 170
Simulation Last State: "Current State: pCustEntry + 28pDTCustLeave + 33pDTFinish + 41pFinish + 42pRCustLeave + 25pTakeAway"

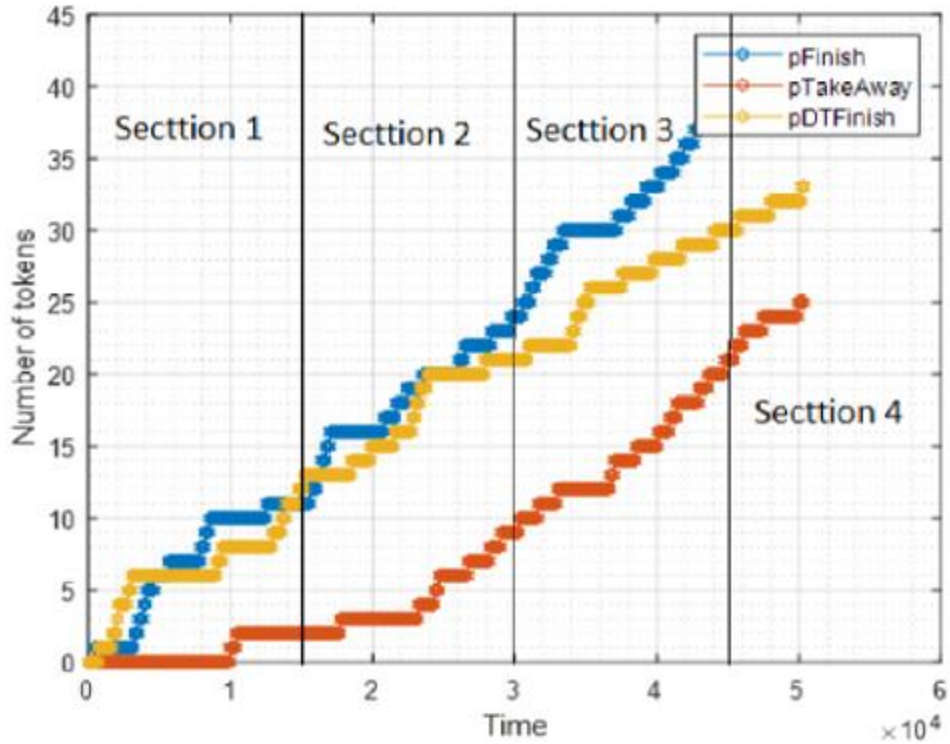

Figure 15: Sample run for 14 hours (Dine In, Take Away & Drive Thru Customers)

This plot represents the single day duration of the restaurant that is fourteen hours on Sunday. It has been generated by running simulation for 50400 Tu. It's X-axis represents Time which is the working duration of the day and Y-axis shows the number of tokens or customers which were entertained during the whole working duration.

Here I am doing analysis of three types of customers namely: Dine Inn, Takeaway and Drive Thru.

Blue line represents Place *pFinish* which shows Dine Inn customers, red line represents Place pTakeAway which shows TakeAway customers while yellow line represents Place *pDTFinish* which shows Drive Thru customers. I have divided the whole daytime in four sections namely: Morning, Afternoon, Evening and Night which are represented by Section 1, Section 2, Section 3 and Section 4 respectively.

Section 1: Highest numbers of customers belong Drive Thru type while lowest is Takeaway type. There is no takeaway customer until mark 1, after that it starts to rise.

Section 2: Dine inn and Drive thru customer types makes most of the customers in numbers. While takeaway customers are lowest in numbers.

Section 3: Here also Dine inn and Drive thru customer types make progress in numbers while takeaway customers being the lowest.

Section 4: Ultimately at dinner time dine inn customers are more in numbers than other two types.

This plot also suggests that dinner time is the busiest while breakfast time is off peak time of the specific day.

4.2.4 Plot run for: 57600 TU
Customer Count: 197
Simulation Last State: **"**Current State: pCustEntry + 31pDTCustLeave + 42pDTFinish + 39pFinish + 45pRCustLeave + 39pTakeAway"

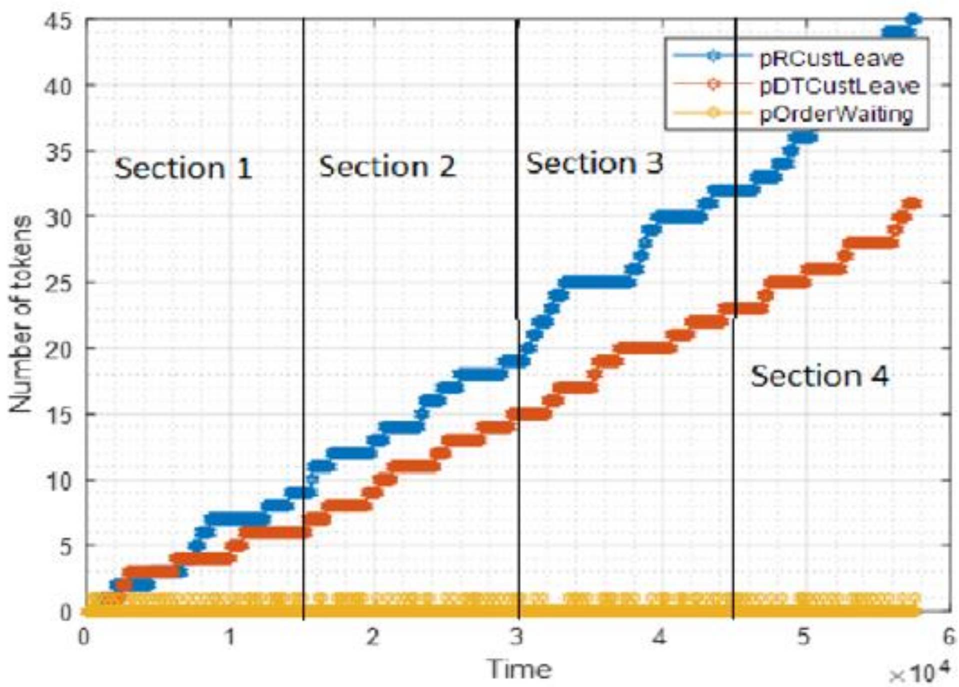

Figure 16: Sample run for 16 hours (Leaving Customers & Waiting Orders)

This plot represents the single day duration of the restaurant that is sixteen hours from Monday until Friday. It has been generated by running simulation for 57600 Tu. It's X-axis represents Time which is the working duration of the day and Y-axis shows the number of tokens or customers which were entertained during the whole working duration.
Customers leave the order queue because of various reasons like somebody forgot his wallet, due to sudden change in plan of eating somewhere else or may be because of any emergency.

Here I am doing analysis of leaving customers who leave order queue like Dine inn customer or takeaway customer which are represented by *pRcustLeave* with a blue line. While Drive Thru customers, who leave the order queue are represented by *pDTCustLeave* with a red line. On the other hand, yellow line represents *pOrderWaiting* which tells us how many orders are in waiting list, pending to go into production at any given time throughout the day.

I have divided the whole daytime in four sections namely: Morning, Afternoon, Evening and Night which are represented by Section 1, Section 2, Section 3 and Section 4 respectively.

Section 1: *pOrderWaiting* remains constant at a value of one because I have controlled it in a way that as soon as an order arrives it go straight into production without any delay to minimize the serving time to the customer and fulfilling customer's satisfaction. While, the number of dine inn and takeaway leaving customers represented by *pRCustLeave* with a blue line is greater in number than the customers who are leaving from drive thru order queue represented by *pDTCustLeave* by red line.

Section 2: *pOrderWaiting* represented by yellow line remains constant at a value of one. Again, the number of dine inn and takeaway leaving customers represented by *pRCustLeave* with a blue line is greater in number than the customers who are leaving from drive thru order queue represented by *pDTCustLeave* by red line.

Section 3: *pOrderWaiting* represented by yellow line remains constant at a value of one. While red and blue line becomes constant for some time between mark 3 to 4, then starts increasing gradually.

Section 4: Again, *pOrderWaiting* represented by yellow line remains constant at a value of one. The number of dine inn and takeaway leaving customers represented by *pRCustLeave* with a blue line is greater in number than the customers who are leaving from drive thru order queue represented by *pDTCustLeave* by red line.

4.2.5 Plot run for: 54000 TU
Customer Count: 175
Simulation Last State: "Current State: pCustEntry + 36pDTCustLeave + 43pDTFinish + 27pFinish + 39pRCustLeave + 29pTakeAway"

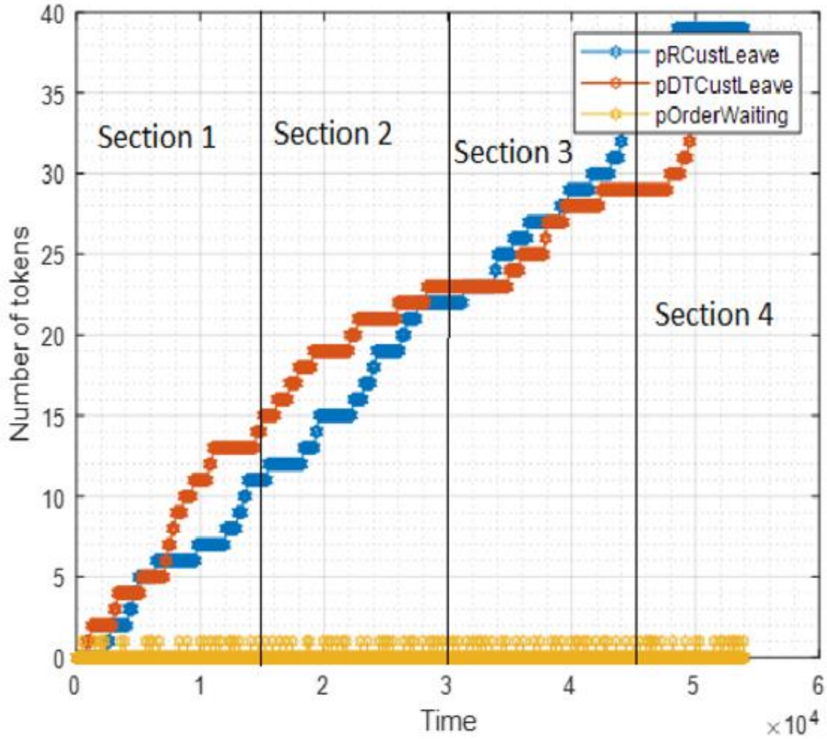

Figure 17: Sample run for 15 hours (Leaving Customers & Waiting Orders)

This plot represents the single day duration of the restaurant that is fifteen hours for Saturday. It has been generated by running simulation for 54000 Tu. It's X-axis represents Time which is the working duration of the day and Y-axis shows the number of tokens or customers which were entertained during the whole working duration.
Customers leave the order queue because of various reasons like somebody forgot his wallet, due to sudden change in plan of eating somewhere else or may be because of any emergency.

Here I am doing analysis of leaving customers who leave order queue like Dine inn customer or takeaway customer which are represented by *pRcustLeave* with a blue line. While Drive Thru customers, who leave the order queue are represented by *pDTCustLeave* with a red line. On the other hand, yellow line represents *pOrderWaiting* which tells us how many orders are in waiting list, pending to go into production at any given time throughout the day.

I have divided the whole daytime in four sections namely: Morning, Afternoon, Evening and Night which are represented by Section 1, Section 2, Section 3 and Section 4 respectively.

Section 1: *pOrderWaiting* remains constant at a value of one because I have controlled it in a way that as soon as an order arrives it go straight into production without any delay to minimize the serving time to the customer and fulfilling customer's satisfaction. From the beginning to the end more customers from drive thru were leaving as compared to the dine inn or takeaway customers.

Section 2: *pOrderWaiting* represented by yellow line remains constant at a value of one. Again, the number of drive thru leaving customers represented by *pDTCustLeave* with a red line is greater in number than the dine inn or takeaway customers represented by *pRCustLeave* with a blue line.

Section 3: *pOrderWaiting* represented by yellow line remains constant at a value of one. In the beginning drive thru leaving customers were in greater numbers but at the end, Dine Inn or takeaway customers are the most who leave the order queue.

Section 4 : *pOrderWaiting* represented by yellow line remains constant at a value of one. Dine Inn or takeaway customers are the most who leave the order queue as compared to drive thru customers.

4.2.6 Plot run for: 50400 TU

Customer Count: 166

Simulation Last State: "Current State: pCustEntry + 31pDTCustLeave + 34pDTFinish + 39pFinish + 27pRCustLeave + 34pTakeAway"

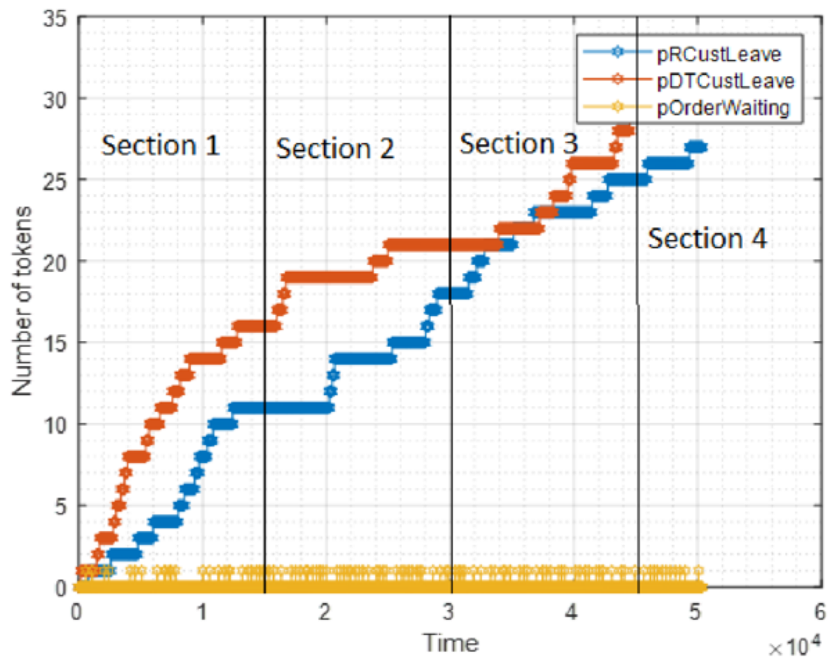

Figure 18: Sample run for 14 hours (Leaving Customers & Waiting Orders)

This plot represents the single day duration of the restaurant that is fourteen hours for Sunday. It has been generated by running simulation for 50400 Tu. It's X-axis represents Time which is the working duration of the day and Y-axis shows the number of tokens or customers which were entertained during the whole working duration.

Customers leave the order queue because of various reasons like somebody forgot his wallet, due to sudden change in plan of eating somewhere else or may be because of any emergency.

Here I am doing analysis of leaving customers who leave order queue like Dine inn customer or takeaway customer which are represented by *pRcustLeave* with a blue line. While Drive Thru customers, who leave the order queue are represented by *pDTCustLeave* with a red line. On the other hand,yellow line represents *pOrderWaiting* which tells us how many orders are in waiting list, pending to go into production at any given time throughout the day.

I have divided the whole daytime in four sections namely: Morning, Afternoon, Evening and Night which are represented by Section 1, Section 2, Section 3 and Section 4 respectively.

Section 1: *pOrderWaiting* remains constant at a value of one because I have controlled it in a way that as soon as an order arrives it go straight into production without any delay to minimize the serving time to the customer and fulfilling customer's satisfaction. From the beginning to the end more customers from drive thru were leaving as compared to the dine inn or takeaway customers.

Section 2: *pOrderWaiting* represented by yellow line remains constant at a value of one. Both, red and blue lines show some stagnancy from mark 1.5 to 3 but at the end drive thru customer were leaving in more numbers from order queue.

Section 3: *pOrderWaiting* represented by yellow line remains constant at a value of one. Both, red and blue lines show some stagnancy from mark 3 to 4.5 but at the end drive thru customer were leaving in more numbers from order queue.

Section 4: *pOrderWaiting* represented by yellow line remains constant at a value of one. In, this section also *pDTCustLeave* line is on the rise, thus having more drive thru customers leaving.

4.3 Data Description:

As described earlier, my project is all about McDonald's Customer Experience which include three major types of customer i.e. Dine In, Take Away and Drive thru. It also possesses two other types for leaving customers like: Dine In/Take Away customers and Drive Thru leaving customers who leaves the order queue.

For my project I have chosen McDonald's restaurant situated at Mariero, Stavanger. Its infrastructure detail is as follows:

Parameter	Value
Automated Ordering Machines (Kiosk Machines)	3(6 Ordering Displays in total)
Delivery Counter	1
Sauce Dispenser	2
Tissue Paper Dispenser	1
Salt and Pepper Dispenser	1
Seating Capacity	60
Toilet	1
Number of Door for customer entry	2

Table 1: Infrastructure Detail

Unfortunately, I could not get the authentic & official data, relatable to my project for this restaurant i.e. number of customers according to my specified types as described above, customers arrival interval information etc.

That is why I opted for customer's probability distribution technique, in which I have assumed the customers probabilities as per my wish to simulate the customer's process. But my model is flexible to changes and in case when real data is available, it can accommodate that data as well and perform the simulations. I ran the simulations with the following assumed customer's probabilities:

Customer Types	Customer's Probabilities
Dine In	0.2
Take Away	0.2
Drive Thru	0.2
RcustLeave (Dine In & Take Away leaving customers)	0.2
DTCustLeave (Drive Thru leaving customers)	0.2

Table 2: Customer Types & Probabilities

Along with the desired customer's probabilities, I also ran simulation by various simulation stoppage times representing restaurant's weekdays and weekend operational timings:

Days	Timings
Mon-Fri	7 AM-11 PM (16 hours/57600TU)
Sat	8 AM-11 PM (15 hours/54000TU)
Sun	9 AM-11 PM (14 hours/50400TU)

Table 3: Simulation Timings

4.4 Statistical Analysis:

In the absence of real data, I could not do much related to statistical analysis but with my assumed data we can get the information on which type of customers are visiting more often in a single day. So that we can focus on them for more custom satisfaction and eradicate the deficiencies in any department, provided if any. In section 4.2 Sample run, I was able to conclude the peak and off-peak hours of the restaurant. This is information is useful for both the customers and the restaurant management. Customers can take advantage by avoiding those rush hours to get their food in less time. While restaurant's management can take benefit on operational level, by keeping in mind that they should scale up their performance in those rush hours for complete customer's satisfaction.

Also, I have analyzed my model with respect to petri net properties which gives us some useful information.

4.4.1 Petri Net Properties Analysis:

Following analysis of my model is done under the concept of petri net properties:

4.4.1.1 Safeness:

All the places in my model are safe at any given time except the following:

1. *pRCustLeave*
2. *pDTCustLeave*
3. *pFinish*
4. *pTakeAway*
5. *pDTFinish*

Because the above 5 mentioned places contain more than one token as they collect the five types of our customers. Hence, I cannot say my petri net model as a whole is safe.

4.4.1.2 Boundedness:

As shown in figure all the places in my model except *pCustEntry* are infinite which means they have duplicate states. So, by the definition of boundedness property of petri net all the places should be bounded to declare a system as bounded, that's why my system is not bounded.

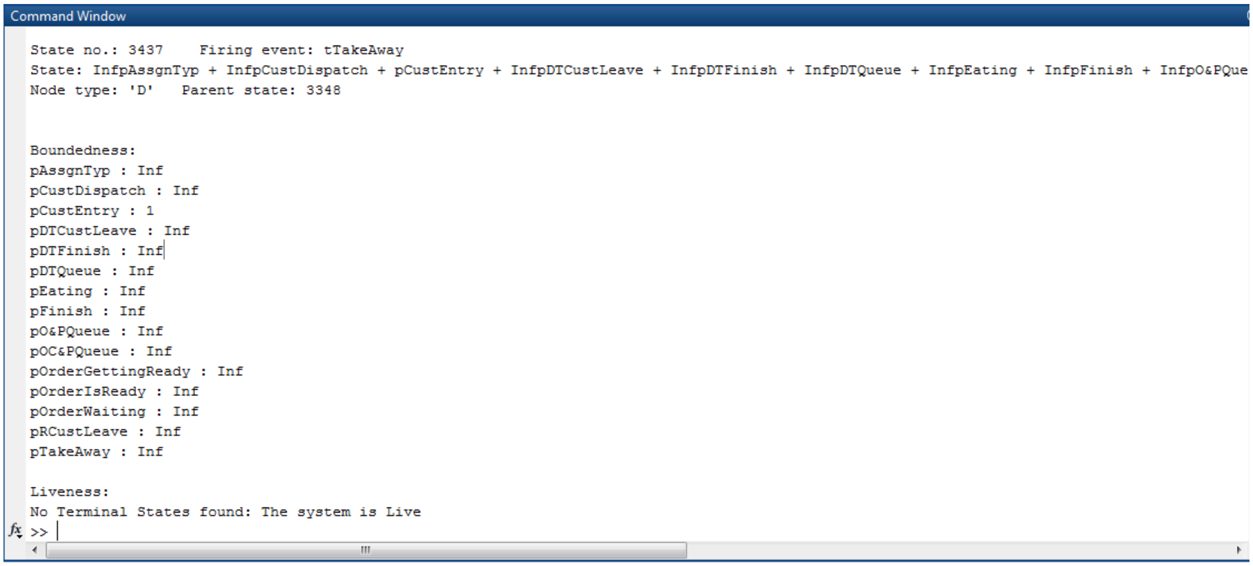

Figure 19: Boundedness

4.4.1.3 Liveness:

Here are five levels of liveness ranging from zero to four and I will describe them in relation to my model in the following:

Level 0: there is no dead transition in my model or no transition which is at level 0

Level 1: all the transitions are at level 1 because they can fire at least once

Level 2: there is only one transition at level 2 i.e., *tCustEntry* which can fire infinitely many times, but not unconditionally because I have allocated a firing time to it

Level 3: there is no transition at level 3 of liveness

Level 4: transitions like *tOrderPreparation* and *tOrderIsReady* are at level 4 because they are enabled/reachable for any kind of customer like Dine Inn, Drive Thru and TakeAway

4.4.1.4 Deadlock:

Each and every transition in my model can be enabled. Thus, they are live and there is no deadlock between them. But as a matter of fact, I have controlled the firing of transitions through pre- processor conditions in COMMON_PRE.m file which depends on the color of token generated which represents type of customer we have to entertain for instance: dine Inn, takeaway or drive-thru customer.

4.5 Reachability Tree (Graphic Output):

The following Reachability Tree has height equal to 15 and an infinite number of duplicate states.

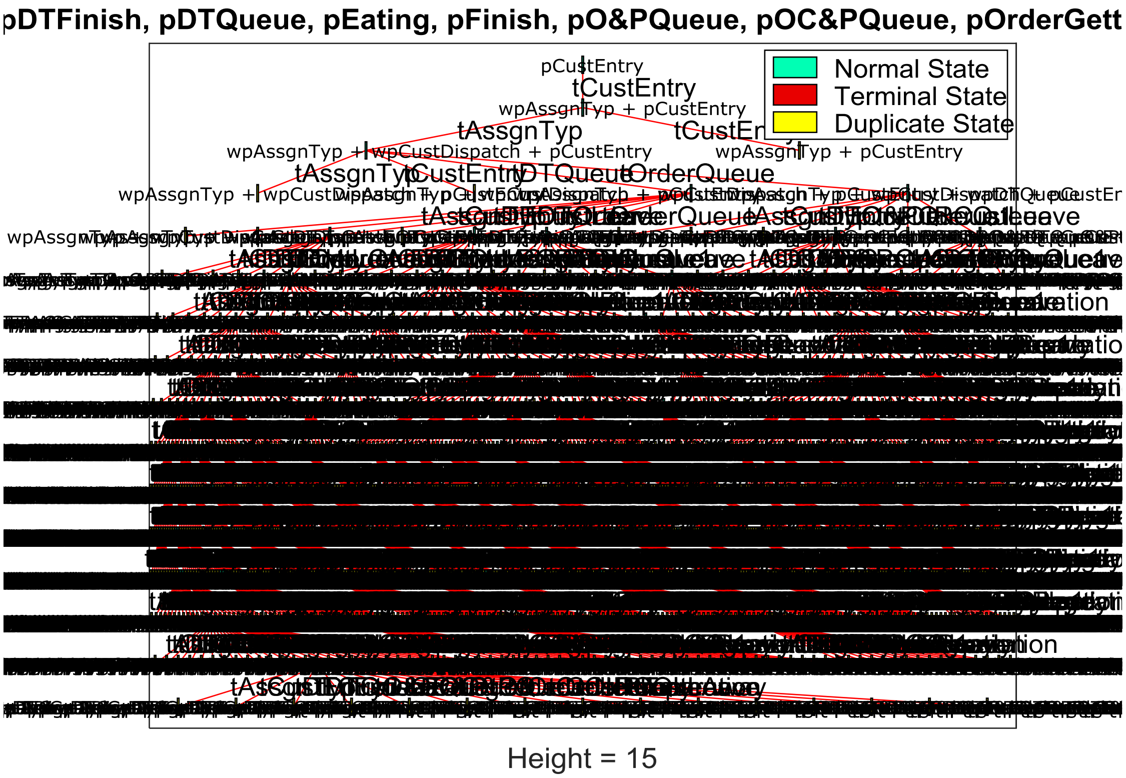

Figure 20: Reachability Tree Output

5.Discussion:

5.1 Originality of Work:

I was unable to get data or similar work done by me whether from the internet, past projects or from any other source regarding McDonald's Customer process simulation.

5.2 Relevant Works:

I have adapted the token generation module from a past project named as "The Drop Zone by Eric Freeman, Ole Tobiesen and Gabriel Nketah". In my analysis part, I used the idea of peak and off-peak hours of my restaurant from another past project named as "Student Canteen Process Simulation by Wei Liao & Long Cui".

5.3 Limitation of my work:

Because of the unavailability of the real data I could not do enough in statistical analysis that is why I had to opt for assumed data approach. Due to the fact of working alone, I some time got short of ideas. Also, I had the challenge of learning petri nets and a new simulator like MATLAB in time a constrained environment.

5.4 Further Work:

Further Work may include the following modules:

1. Another customer type can be included like Bulk order customer type which is different events or gatherings.
2. Online food delivery type can also be modelled
3. Module can be added for analyzing the performance of human resource
4. Stocks and inventory management module can also be useful for the restaurant management.
5. Module for analyzing the most favorite meal or menu among the customers

5.5 Learning Experience:

It was a great experience of learning a new modelling technique like petri nets using GPenSIM and MATLAB during limited amount of time. It was challenging for me to complete my project in time but at the end I achieved my goal.

References:

1. http://www.davidrajuh.net/gpensim/v9/GPenSIM_v9_User_Manual.pdf
2. Past Projects which are made available in CANVAS by professor Reggie Davidrajuh
3. https://www.google.no/search?biw=1366&bih=657&tbm=isch&sa=1&ei=hq7pW4CVM_KprgSL74-gBg&q=mcdonalds+mariero&oq=mcdonalds+marier&gs_l=img.3.0.0i30k1j0i24k1.20356.26580.0.27942.7.7.0.0.0.0.167.616.6j1.7.0....0...1c.1.64.img..0.7.610...0j35i39k1j0i67k1j0i10k1j0i8i30k1j0i10i24k1.0.PyOhR39eiUM#imgrc=gPy3QdrIfDSQQM:

Appendix-A:

A1: Installation Guide:

1. First install MATLAB and GPenSIM.
2. Unzip the file named as “McDonald’sCustomer’sExperience.rar”

A2: User Manual:

1. This file will have a folder named as “McDonald_Project”.
2. Open this folder in order to run simuation.
3.Open the main simulation file named as “McDonalds.m”.
4. You can make changes as per requirement to the simulation stoppage time and to the customer’s probabilities as shown in Code Snippet 17.
5. After making changes, run “McDonalds.m”.

A3: Complete Code:

A3.1 Main Simulation File (MSF):

```
clear all; clc;

global global_info
%16 hours working shift of the restaurant is represented by 576000 seconds
global_info.STOP_AT = 57600;
%represents 5 types of customers in the form of colors
%'RCustLeave' includes DineInn & TakeAway Leaving customers while
%'DTCustLeave' includes DriveThru Leaving customers
global_info.CustomerType = {'DineInn', 'TakeAway',
'DriveThru','RCustLeave','DTCustLeave'};
%It is the probability of 5 types of customers
global_info.CustomerProbability = [0.2 0.2 0.2 0.2 0.2 ];
% time and counter values are used for the token generation
time = 0;
counter = 0;
global_info.TokenGeneration_Times = zeros(1, 20000); %An array having token
generation times
% For the generation of customers untill the specified time
while lt(time , 10000)
    counter = counter + 1;
    time = time + normrnd(410, 200);
    global_info.TokenGeneration_Times(counter) = time;
end

pns = pnstruct('McDonalds_pdf');
dyn.m0 = {'pCustEntry', 1};
dyn.ft = {'tCustEntry', 5,...
          'tO&PQueue1',10,...
          'tOrderPreparation',10,...
          'tOrderIsReady',10,...
          'tOrderCollection',10,...
          'tFinishEating',10,...
          'tTakeAway',10,...
          'tAssgnTyp',10,...
          'tOrderQueue',10,...
          'tDTQueue',10,...
          'tOrder',10,...
         'tOC&PQueue',10,...
         'tOC&P',10,...
         'tRCustLeave',10,...
         'tDTCustLeave',10};

pni = initialdynamics(pns, dyn);
sim = gpensim(pni);
  prnss(sim);
% cotree(pni,1,1);
prnfinalcolors(sim);
% prnschedule(sim);
%plotp(sim,{'pFinish','pTakeAway','pDTFinish'})
%plotp(sim, {'pRCustLeave','pDTCustLeave','pOrderWaiting'});
disp(['Total number of customers: ', num2str(counter)]);
```

A3.2 PDF File:

```
function [png] = McDonalds_pdf()
png.PN_name = 'McDonalds_pdf';
png.set_of_Ps =
{'pCustEntry','pO&Pqueue','pOrderWaiting','pOrderGettingReady','pOrderIsReady
','pEating','pFinish','pTakeAway',…

'pAssgnTyp','pCustDispatch','pDTQueue','pOC&Pqueue','pDTFinish','pRCustLeave'
,'pDTCustLeave'};

png.set_of_Ts =
{'tCustEntry','tO&Pqueue1','tOrderPreparation','tOrderIsReady','tOrderCollect
ion','tFinishEating','tTakeAway',…

'tAssgnTyp','tOrderQueue','tDTQueue','tOrder','tOC&Pqueue','tOC&P','tRCustLea
ve','tDTCustLeave'};

png.set_of_As = {    'pCustEntry','tCustEntry',1, …
                     'tCustEntry','pCustEntry',1,…
                     'tCustEntry','pAssgnTyp',1,…
                     'pAssgnTyp','tAssgnTyp',1,…
                     'tAssgnTyp','pCustDispatch',1,…
                     'pCustDispatch','tOrderQueue',1,…
                     'tOrderQueue','pO&Pqueue',1,…
                     'pO&Pqueue','tO&Pqueue1',1,…
                     'tO&Pqueue1','pOrderWaiting',1,…
                     'pO&Pqueue','tRCustLeave',1,…
                     'tRCustLeave','pRCustLeave',1,…
                    'pCustDispatch','tDTQueue',1,…
                     'tDTQueue','pDTQueue',1,…
                     'pDTQueue','tOrder',1,…
                     'tOrder','pOrderWaiting',1,…
                     'pDTQueue','tDTCustLeave',1,…
                     'tDTCustLeave','pDTCustLeave',1,…
                      'pOrderWaiting','tOrderPreparation',1,…
                      'tOrderPreparation','pOrderGettingReady',1,…
                      'pOrderGettingReady','tOrderIsReady',1,…
                      'tOrderIsReady','pOrderIsReady',1,…
                      'pOrderIsReady','tOC&Pqueue',1,…
                      'tOC&Pqueue','pOC&Pqueue',1,…
                      'pOC&Pqueue','tOC&P',1,…
                      'tOC&P','pDTFinish',1,…
                      'pOrderIsReady','tOrderCollection',1,…
                      'tOrderCollection','pEating',1,…
                      'pEating','tTakeAway',1,…
                    'tTakeAway','pTakeAway',1,…
                      'pEating','tFinishEating',1,…
                      'tFinishEating','pFinish',1,…
                      };
```

A3.3 COMMON_PRE FILE:

```
function [fire, transition] = COMMON_PRE(transition)
global global_info;

%this transition is for creating a new token in the form of a customer
if (strcmpi(transition.name, 'tCustEntry'))
    % Don't let the transition fire if the array is empty
    if isempty(global_info.TokenGeneration_Times)
        fire = 0;
        return;
    end;
    cTime = current_time(); %to get the current time
    genrationTime = global_info.TokenGeneration_Times(1);
    % Don't let the transition fire if the remaining generation times in the
array are equal to 0
    if eq(genrationTime , 0)
        fire = 0;
        global_info.TokenGeneration_Times = [];
        return;
    end;
    % Don't let the transition fire if current time is lower than the
generation time
    if lt(cTime , genrationTime)
        fire = 0;
        return;
    end;
    % It is time to fire
    if ge(length(global_info.TokenGeneration_Times), 2)
        global_info.TokenGeneration_Times =
global_info.TokenGeneration_Times(2:end);
    else
        global_info.TokenGeneration_Times = [];
    end;
    fire = 1;

    %this transition is for assigning a random color to the customer/token
elseif(strcmpi(transition.name, 'tAssgnTyp'))

 %Function for getting cumulative distribution of customers
    cumDist = [0 cumsum(global_info.CustomerProbability)];
    %for the random selection of customers in relation to the given...
    %customer probability
    select = rand();
    greater = find(cumDist >= select==1,1);
    transition.new_color = global_info.CustomerType{greater-1};
    transition.override = true;
    fire=1;

    %this transition is for selecting a token if it's color is one of these
i.e., 'DineInn', 'TakeAway','RCustLeave'
elseif(strcmpi(transition.name, 'tOrderQueue'))
```

```
    tokID1 = tokenAnyColor('pCustDispatch',1, {'DineInn',
'TakeAway','RCustLeave'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
  %this transition is for selecting a token if it's color is one of these
i.e., 'DriveThru','DTCustLeave'
elseif(strcmpi(transition.name, 'tDTQueue'))
    tokID1 = tokenAnyColor('pCustDispatch',1, {'DriveThru','DTCustLeave'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
   %this transition is for selecting a token if it's color is exactly equal
to 'DriveThru'
elseif(strcmpi(transition.name, 'tOrder'))
    tokID1 = tokenEXColor('pDTQueue',1, {'DriveThru'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
    %this transition is for selecting a token if it's color is exactly...
    %equal to 'DTCustLeave'
    elseif(strcmpi(transition.name, 'tDTCustLeave'))
    tokID1 = tokenEXColor('pDTQueue',1, {'DTCustLeave'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
    %this transition is for selecting a token if it's color is one of these
    %i.e., 'DineInn', 'TakeAway'
elseif(strcmpi(transition.name, 'tO&PQueue1'))
   tokID1 = tokenAnyColor('pO&PQueue',1, {'DineInn', 'TakeAway'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
    %this transition is for selecting a token if it's color is exactly...
    %equal to 'RCustLeave'
    elseif(strcmpi(transition.name, 'tRCustLeave'))
   tokID1 = tokenEXColor('pO&PQueue',1, {'RCustLeave'});
    transition.selected_tokens = tokID1;
    fire = tokID1;

elseif(strcmpi(transition.name, 'tOrderPreparation'))
    fire=1;

elseif(strcmpi(transition.name, 'tOrderIsReady'))
    fire=1;
    %this transition excludes tokens with color 'DriveThru' and selects
    %tokens only with colors 'DineInn', 'TakeAway'
elseif(strcmpi(transition.name, 'tOrderCollection'))
     tokID1 = tokenWOEXColor('pOrderIsReady',1, {'DriveThru'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
    %this transition selects token with exact color i.e., 'DriveThru'
elseif(strcmpi(transition.name, 'tOC&PQueue'))
     tokID1 = tokenEXColor('pOrderIsReady',1, {'DriveThru'});
    transition.selected_tokens = tokID1;
    fire = tokID1;

elseif(strcmpi(transition.name, 'tOC&P'))
    fire=1;
   %this transition selects token with exact color i.e., 'DineInn'
elseif(strcmpi(transition.name, 'tFinishEating'))
```

```
   tokID1 = tokenEXColor('pEating',1, {'DineInn'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
    %this transition selects token with exact color i.e., 'TakeAway'
elseif(strcmpi(transition.name, 'tTakeAway'))
     tokID1 = tokenEXColor('pEating',1, {'TakeAway'});
    transition.selected_tokens = tokID1;
    fire = tokID1;
end;
```