Simulation of Food Restaurant Using Colored Petri Nets

Lecture Zinah Salih Hussein
computer science , Al_ Mustansiriyah University, Baghdad_ Iraq
E-mail zezefun@yahoo.com

Abstract

a food restaurant approach is proposed in this paper. The proposed approach is based on moving the customer to request meal and how the sensitivity for changes by applying a new tool for modeling and simulating works of the restaurant.

Colored Petri nets (CPNs) are applied in modeling specific characteristics of restaurant system, and suppose the customer moving through the restaurant by a token firing. Furthermore GPenSIM (General Purpose Petri Net Simulator) is used as a new tool for modeling and simulating of discrete event system. GPenSIM is realized as a toolbox for the MATLAB platform, so that diverse toolboxes that available in the MATLAB environment.

Keywords: Customer, restaurant, food, Colored Petri Nets (CPNs), GPenSIM (General purpose Petri Net Simulator).

محاكاة عمل المطاعم باستخدام شبكات ألبتري الملونة

م.م. زينة صالح حسين
الجامعة المستنصرية / كلية العلوم / قسم الحاسوب.

الخلاصة :

في هذا البحث تم اقتراح نهجا جديدا لمعالجة المشاكل التي تعاني منها المطاعم ذات الوجبات السريعة. ويستند النهج المقترح على انتقال الزبون لطلب الوجبة وكيفية التحسس بالتغيرات التي تحصل من خلال تطبيق أداة جديدة لنمذجة ومحاكاة العمل في المطاعم. وقد تم أتباع نوع معين من الشبكات هو (CPNs) لنمذجة الخصائص الخاصة بنظام المطاعم، على افتراض أن الزبون يتحرك في المطعم بشكل قيم رمزية تطبق خلال الشبكة .وعلاوة على ذلك تم استخدام GPenSIM وهي احد أدوات العمل التي وفرتها بيئة MATLABوالتي تحاكي العمل بنظام الأحداث المتقطعة .

1. Introduction:

A Petri net is a particular kind of bipartite directed graphs populated by three types of objects. These objects are *places*, *transitions*, and *directed arcs*. Directed arcs connect places to transitions or transitions to places. In its simplest form, a Petri net can be represented by a transition together with an input place and an output place. This elementary net may be used to represent various aspects of the modeled systems. For example, a transition and its input place and output place can be used to represent a data processing event, its input data and output data, respectively, in a data processing system. In order to study the dynamic behavior of a Petri net modeled system in terms of its states and state changes, each place may potentially hold either none or a positive number of *tokens*. Tokens are a primitive concept for Petri nets in addition to places and transitions. The presence or absence of a token in a place can indicate whether a condition associated with this place is true or false, for instance [1].

A. structural components of Petri nets

This section describes the component of Petri net applied in a restaurant. [4]

- A place represents the start or completion of an activity. They are represented by circles . The customer transfer through the following states of readiness: customer enter to Corner sandwiches, Corner soup, Corner bagel.
- A transition represents activity, which is a time/resource consuming part of a project. They are represented by bars.
- An arc connects places and transitions. It is not allowed to connect two places or two transitions. The arcs are either from a transition to a place or from a place to transition.
- Token represent resources of the operation. These are represented by black dot. In this case the tokens are the customers.

B. Relational Mathematical Definitions

A Petri net is formally defined as a 5-tuple $N = (P, T, I, O, M0)$where [1]:

1. $P = \{p1,\ p2, \ldots., p\ m\ \}$ is a finite set of places;

2. $T = \{t1, t2, \ldots., t\ n\}$ is a finite set of transitions;

3. I: P × T → N is an input function that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

4. O: T × P→ N is an output function that defines directed arcs from transitions to places;

5. $M0$: $P \rightarrow N$ is the *initial marking*.

A marking in a Petri net is an assignment of tokens (customers) to the places of a Petri net. Tokens reside in the places of a Petri net. The number and position of tokens may change during the execution of a Petri net. The tokens are used to define the execution of a Petri net.

1.2 Characterizes of Petri Nets:

This section identifies several PN constructs that represent particular translation of customer through the restaurant: [5]

A. Sequential execution

Transition (t_2) can fire only after the firing of (t_1).This imposes a precedence constraint that is typical of some construction activities. The construct can also be used to model causal relationships between activities, [8] see **Figure (1).**

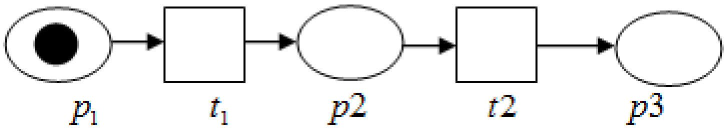

Fig .(1) ***Sequential execution***

B. Conflict

Petri nets are most suitable for describing the conflicts [8]. Conflict indeed makes a choice determining with Petri net. **Figure. (2)** indicates that when transition t_1 is firing, a token will be located in the place P and transitions t_2 will be enabled. When one of them is fired, it will remove the token from place P and thus another transition will be disabled.[5]

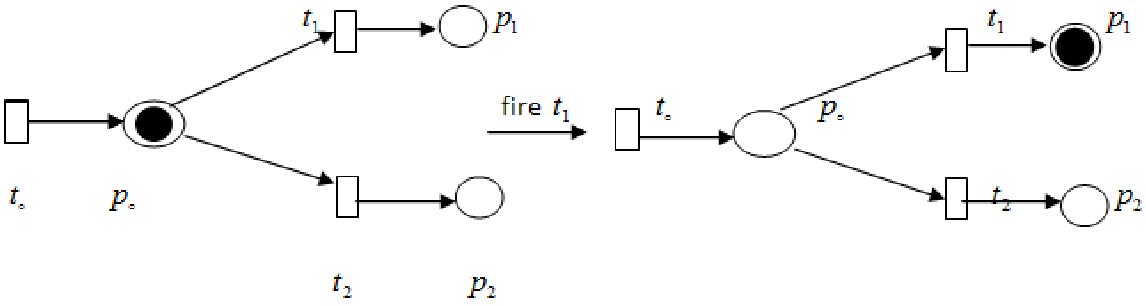

Fig .(2) conflict situation in Petri nets

C. Concurrency

after firing the transition t_0 then the transitions t_1, t_2 *and* t_3 are concurrent. This construct allows to represent constructive activities that proceed in parallel [8], as shown in **Figure.3**

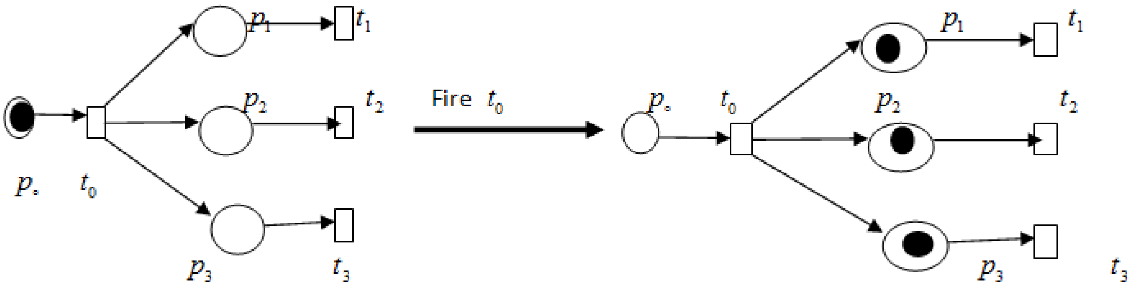

Fig .(3) concurrency situation in Petri net

1.3 Colored Petri Nets

Colored Petri Nets (CPNs) differ from PNs in one significant respect; tokens are not simply blank markers, but have data associated with them. A token's color is a schema or type specification. Places are then sets of tuples, called **multi-set**. Arcs specify the schema they carry, and can also Specify basic Boolean conditions. Specially, arcs exiting and entering a place may have an associated function which determines what multi-set elements are to be removed or deposited. Simple Boolean expressions, called guards, are associated with the transitions, and enforce some constraints on tuple elements. CPNs are formally equivalent to traditional PNs؛ however, the richer notation makes it possible to model in tractions in CPNs where it would be impractical to do so with PNs **[6].** component of CPNs:

- The states of a CPNs are represented by means of **places** (which are drawn as circles). write the names of the places beside the circles. Each place has an associated **type** determining the kind of data that the place may contain.**[3]**
- During the execution of a CPNs each place will contain a varying number of **tokens**. Each of these tokens carries a data value that belongs to the type associated with the place. In general, several tokens may have the same token value, and then we have a **multi-set** of token values, such as represent token (customers). A state of a CPN is called a **marking**. It consists of a number of tokens positioned on the individual places.[3]
- The actions of a CPNs are represented by means of **transitions** (which are drawn as rectangles). An incoming arc indicates that the transition may remove tokens from the corresponding place while an outgoing arc indicates that the transition may add tokens. **[3]**

1.4 GPenSIM (General purpose Petri Net Simulator)

GPenSIM is a new tool for modeling and simulating of discrete event systems, and written in MATLAB language, allows seamless integration with the other toolboxes that also available in the MATLAB environment. GPenSIM Programming in MATLAB Language is also extremely easy as the language resembles the BASIC language. The reasons for building a new simulator are (flexible, extensible, ease of use). A model of a discrete event system developed with GPenSIM consists of a number of files. The main simulation file (MSF) is the file that will be run directly by the MATLAB platform. In addition to the main simulation file, there will be one or more Petri net definition files (PDFs); definition of a Petri net graph (static details) is given in the Petri net Definition File. There may be a number of PDFs, if the Petri net model is divided into many modules, and each module is defined in a separate PDF. While the Petri net definition file has the static details, the main simulation file contains the dynamic information (such as initial tokens in places, firing times of transitions) of the Petri net. In addition to these files (main simulation file and Petri net definition files), there can be a number of transition definition files (TDFs) too. A transition definition file consists of additional conditions that determine whether an enabled transition can fire or not. The additional conditions are called „user defined condition‟ in GPenSIM terminology, whereas in some other literature (e.g. Colored Petri Net (CPN)) it is referred to as „guard-functions‟). There can be a separate transition definition file for each transition in a Petri net model.[5]

The main objective in this paper is to propose an approach to capable of reducing the complexity and modeling time of multi-transitions in restaurants system by using Colored Petri nets.

2. Related Works

Kurt Jensen at 1996 discussed the concept of Colored Petri Nets (CP-nets or CPN) which has been driven by the desire to develop language model at the same time theoretically well-founded and versatile enough to be used in practice for systems of the size and complexity found in typical industrial projects. The strength of Petri nets with the strength of programming languages. Petri nets provide the primitives for describing synchronization of concurrent processes, while programming languages provide the primitives for definition of data types and manipulation of their data values.[3]

Mathias Dharmawirya at 2011 discussed the concept Restaurants would avoid losing their customers due to a long wait on the line. Some restaurants initially provide more waiting chairs than they actually need to put them in the safe side, and reducing the chairs as the time goes on safe space. However, waiting chairs alone would not solve a problem when customers withdraw and go to the competitor’s door; the service time may need to be improved.[2]

Sara A. Curin at 2005 Used the standard simulation study steps, several service scenarios modeling and evaluating were based on customer system time. To reduce customer wait times and, therefore, serve more customers per hour, it is recommended that Tim Hortons operate with five servers. A five-person setup with three cashiers, a soup server, and a sandwich server could reduce customer system time by over two minutes per customer.[7]

3. System Description

The restaurant works twelve hours during the day, meaning that there are two groups working alternately in order to keep the doors of restaurant open daily to provide the services needed by customers. These restaurants are characterized by small size and lack of staff, but at the same time the speed of providing service to customers, as it there is only one cashier records request and gives to customer a paper for demand. As a single server for foods meal and another server is responsible for drinks.

3.1 Acknowledging with the Proposing System

The proposed system is to simulate the restaurant model by using colored tokens and transition the customer from one place to another, when the customer enter to restaurant and demand from cashier card to give the server about the meal of food such as (soup, sandwich, bagel,…) or drinks(coffee, soft drink, …). As the first customer's arrival to the restaurant receive a service first which based on the principle FCFS (First Come First Service), But if the customer's request for the meal by phone without going to the restaurant or he take request directly without stay will be delivered. The Petri net is shown at the initial marking, for which the server (cashier) are receiving demand from customer. as **Figure (4)** summarize the restaurant system by a flow chart.

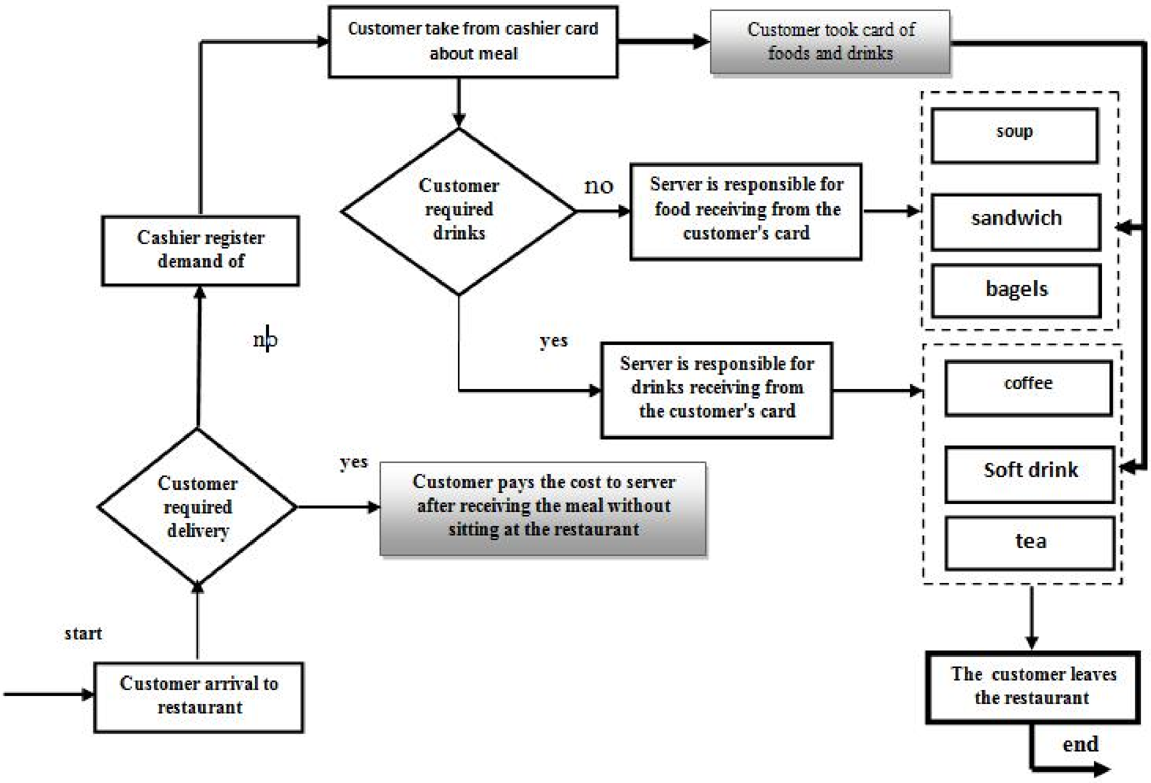
Customer take from cashier card about meal
Customer took card of foods and drinks
Customer required drinks
no
Server is responsible for food receiving from the customer's card
soup
sandwich
bagels
Cashier register demand of
no
yes
Server is responsible for drinks receiving from the customer's card
coffee
Soft drink
tea
yes
Customer required delivery
Customer pays the cost to server after receiving the meal without sitting at the restaurant
start
Customer arrival to restaurant
The customer leaves the restaurant
end

Fig .(4) Flow chart of proposed restaurant system

3.2 Suggested Model for Restaurant System Using Colored Petri Net:

Petri net system of production system studied is indicated in Fig. 5. the model includes **13** places and **12** transitions. Details of each place and transition have indicated in **Table (1 and 2).**

Table (1) Properties of Places

states	Place name	Descriptions
1	P0	customer arrival to restaurant
2	P1	cashier register demand of customer
3	P2	customer take from cashier card of meal
4	P3	server of food receiving from customer card
5	P4	customer required soup
6	P5	customer required sandwiches
7	p6	customer required bagels
8	P7	server of drinks receiving from customer card
9	P8	customer required coffee
10	P9	customer required soft drink
11	P10	customer leave from restaurant
12	P11	customer required from server delivery
13	P12	customer depart from restaurant directly without sitting

Table (2) Properties of Transition

states	Places Name	descriptions
1	t0	customer arrival to restaurant
2	t1	customer access to cashier
3	t2	customer toward to server for food
4	t3	customer toward to soup place
5	t4	customer toward to sandwiches place
6	t5	customer toward to bagels place
7	t6	customer depart from food department
8	t7	customer arrival to server for drinks
9	t8	customer toward to coffee place
10	t9	customer depart from drink department
11	t10	customer arrival to need delivery meal & won’t to stay and eat at the restaurant
12	t11	customer depart leave from restaurant

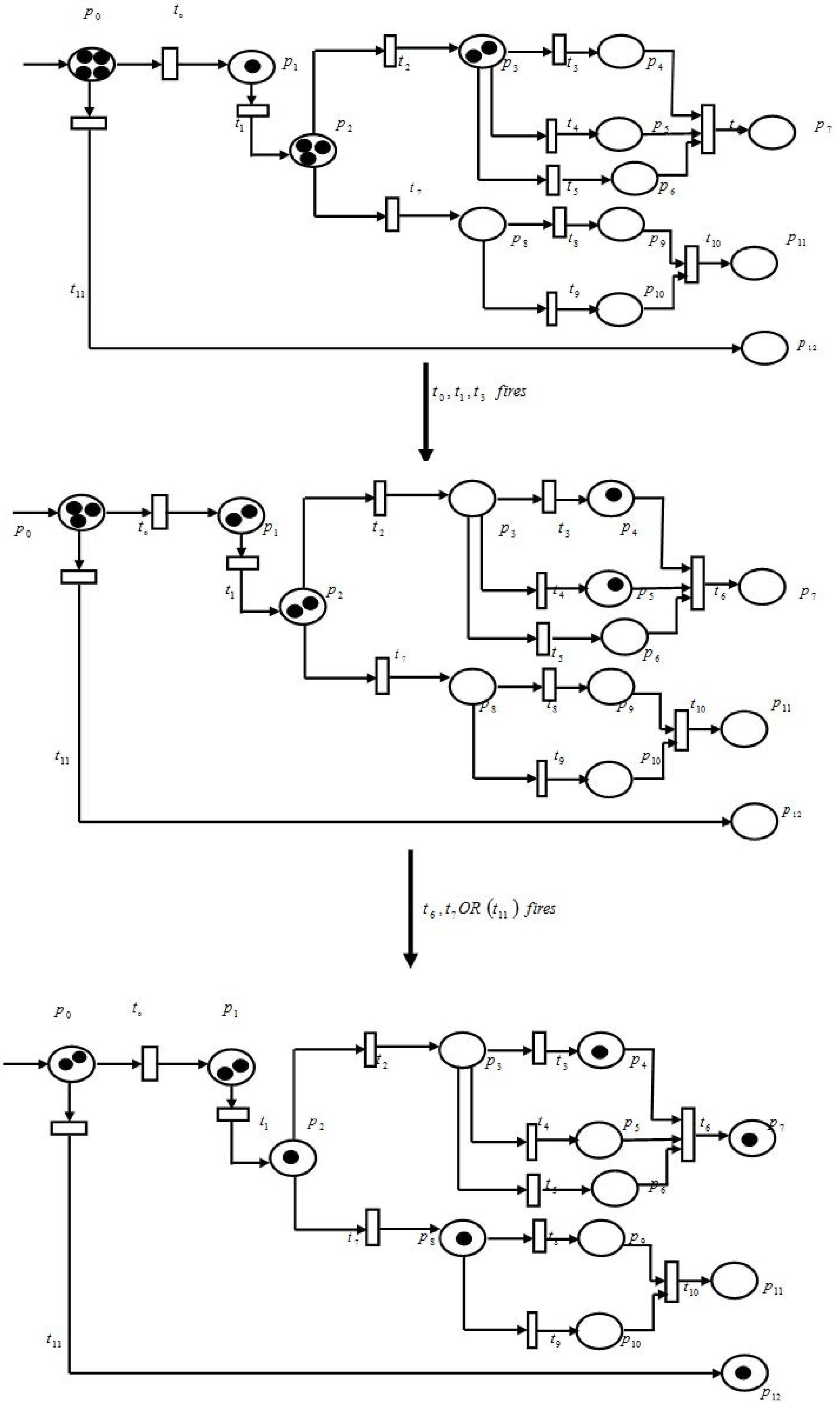

Fig .(5) Color Petri net for restaurant system

4. Methodology for Simulation

This section describes the **function of ‘gpensim’** by defining two parts **first**: static part of Petri net graph have (number of places, number of transitions, number of arcs to connect between element). As **incidence_ matrix** is represent the matrix that depicts how the places and transitions are connected together.

The Color Petri net graph showed in **Figure (5)** has 13 places & 12 transitions:

```
% Example: Restaurant Example
 % file: ’Restaurant_def.m’
 % this file defines the restaurant Petri net graph
Function [PN_name, set_of_places, set_of_trans, set_of_arcs]= restaurant_def (global info)
 PN_name = ' restaurant Petri Net';
 set_of_places = {''P-0', P-1', 'P-2', 'P-3', 'P-4', 'P-5', 'P-6', 'P-7', 'P-8', 'P-9', 'P-10', 'P-11', 'P-12'};
  set_of_trans = {'T-0', 'T-1', 'T-2', 'T-3', 'T-4', 'T-5', 'T-6', 'T-7', 'T-8', 'T-9', 'T-10', 'T-11'};
 set_of_arcs = {‘p0_t1’, 1, ‘p3_t3’, 1, ‘p5_t5’, 1,’ p6_t6’, 1,’p10_ t10’, 1, …]
 incidence_ matrix: [1   1   0  0  0   0    0  0    0   0     0  0  20]
```

Second: after creating Petri net model, simulations can be done.

```
>> cpng = ColorPetri net graph ('restaurant_def');
```

Furthermore, the dynamic part has an initial marks on the places and the firing times of the transition to be assigned, and define by two informations into a packet (e.g. ‘dynamic_info’ in this example) and then pass this packet to function ‘**gpensim**’.

```
  >> dynamic_info.initial_markings = {'P0',20};
 >> dynamic_info.firing_times  = {'T-0', 2, 'T-2', 5, 'T-6’, 10, 'T-7’, 15, 'T-11’, 20};
```

Finally : the function “gpensim” can simulated by:

```
> > Sim_Results = gpensim(cpng, dynamic_info);
```

The argument of Sim _ Results is a structure for the simulation results, and the simulation results can easily to represent by applying the function of ‘print_ state’ and drawn by using the function ‘plot’ .

```
>> print_ state (Sim _ Results);
>> plot (Sim_Results, {'P-0', 'P-1', 'P-2', 'P-3', 'P-4', 'P-5', 'P-6', 'P-7', 'P-8', 'P-9', 'P-10', 'P-11', 'P-12'})
```

Note: The next statement describes the number of tokens (customers) firing in each state. In **state1** $P0$ =19 (No. of token(customer) arrival to restaurant) after **(t0)** fired transition then $P1$ = 1 (one token(customer)in cashier to demand meal). In **state2** $P0$ =16(No. of token(customer)arrival to restaurant), after **(t1)** fired transition then $P1$ = 1 (one token(customer) in cashier to demand meal), $P2$ =2 (two token(customer) takes from cashier card), p3=0 (no token(customer) firing), and **(t3)** is firing transition then p4=1(one token(customer) required soup). when continues to firing of tokens (customers) and transitions until access to **state10** $P0$ =1(one token(customer) arrival to restaurant), $P1$ = 1 (one token(customer) in cashier to demand meal), $P2$ =1 (one token(customer) take from cashier card), p3 and p4=0 (no token (customer) firing), p5=1 (one token(customer) required sand wishes), ect.

The output of print_ state is given below:

```
  State: 0 (Initial State)
                              p1 p2 p3 p4 p5 p6   p7 p8 p9 p10 p11 p12  p0
                              0  0   0   0  0   0  0   0  0  0    0    0     20
 At time: 0                   Enabled transitions are: t0
 At time: 0                   Firing transitions are:  t 0
 Time: 5
State: 1
     Fired Transition: t0, t1
     Current State                 p1 p2 p3 p4 p5 p6   p7 p8   p9 p10 p11 p12  p0
                                   1   0  0  0  0  0  0   0  0   0   0    0    0    1 9
  At time: 5                          Enabled transitions are: t0, t1
  At time: 5                          Firing transitions are: t1
  Time: 10
  State: 2
  Fired Transition: t1, t3
  Current State:                      p1 p2 p3 p4 p5 p6   p7 p8   p9 p10 p11 p12  p0
                                      1   2   0  1  0   0   0  0   0   0    0    0    16
  At time: 10                        Enabled transitions are:  t1, t3
  At time: 10                        Firing transitions are:  t3
  ………………….
  ………………..
 Time: 25
 State: 10
  Fired Transition: t6, t7, t11
  Current State:                    p1 p2 p3 p4 p5 p6   p7 p8   p9 p10 p11 p12   p0
                                    1  1   0  0   1  0   2   1   1   1    0    1     1
 Enabled transitions are: t6, t7, t11       ,          Firing transitions are: t7, t11
```

The above statement will plot how some tokens in the places fire by applying **incidence_ matrix** in **state 2** see that in **Figure (6)** and in the **state 10** illustrated in **Figure (7):**

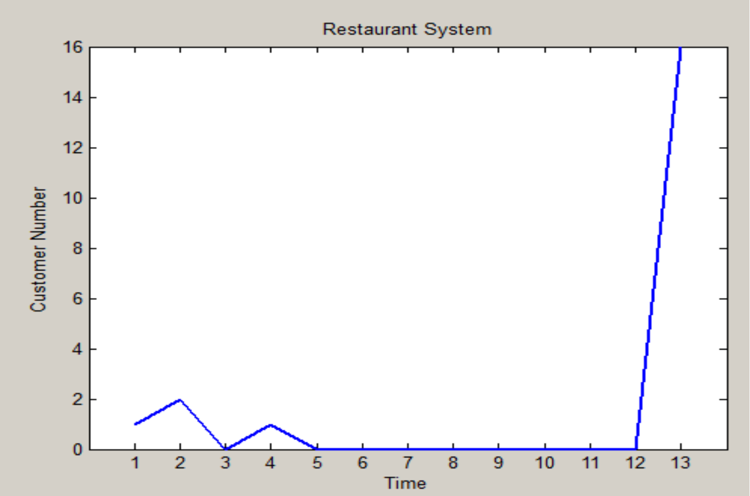

Fig .(6) some token fired

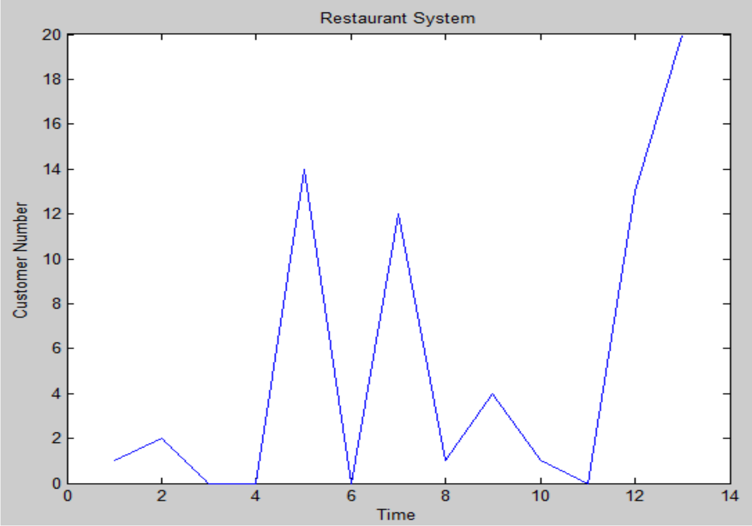

Fig .(7) several tokens fired

5. Conclusions:

The objective of this system is to apply PNs design to access optimum restaurant system. The CPNs used to represent the customer transfer through the restaurant because its flexible compare with the other models, especially suppose the customers number are increasing caused to more tokens are firing. In addition the CPNs reducing space to represent proposal system compared with other methods.

References:

1. **Jiacun Wang, "Petri Nets for Dynamic Event-Driven System Modeling", Monmouth University, West Long Branch, NJ 07764, jwang@monmouth.edu, 2008.**
2. **Mathias Dharmawirya, Erwin Adi, "Case Study for Restaurant Queuing Model", Binus International – Binus University, *IPEDR vol.6, Bali,* Indonesia, 2011.**
3. **Kurt Jensen, "An Introduction to the Practical Use of Colored Petri Nets", University of Aarhus Ny Munkegade, Bldg. 540, DK-8000 Aarhus C, Denmark, E-mail: kjensen@daimi.aau.dk , WWW: http://www.daimi.aau.dk/~kjensen , 1996.**
4. **S. Kumanan and K. Raja, "Modeling and Simulation of Projects with Petri Nets", National Institute of Technology, Tiruchirappalli, India, ISSN 1546-9239, 2008.**
5. **Reggie Davidrajuh, "GPenSIM (General Purpose Petri Net Simulator)", University of Stavanger, Norway Email: reggie.davidrajuh@uis.no, 2010.**
6. **R. Scott Cost, Ye Chen, Tim Finin, "Modeling Agent Conversations with Colored Petri Nets", University of Maryland Baltimore County Baltimore, Maryland cost@acm.org, yechen, nin, jklabrou, ypeng@cs.umbc.edu , 2000.**
7. **Sara A. Curin, Jeremy S. Vosko, "Reducing Service Time at a Busy Fast Food Restaurant on Campus", The University of Michigan, Ann Arbor, MI 48109, U.S.A., 2005.**
8. **Dr. A. Venkateswarlu, " A Study of Petri Nets Modeling, Analysis and Simulation", Control Dynamics and Analysis Division , ISRO Satellite Centre, Bangalore – 560 017, 2006.**