Petri Net Model and Max-Plus Algebra in Outpatient Care at Solo Peduli Clinic, Surakarta

Ghooa Phateena Taqie Haneefa¹, Siswanto²

¹,²) Department of Mathematics, Universitas Sebelas Maret, Indonesia

Corresponding email: ghoodapth@student.uns.ac.id

Abstract. Patients come to SoloPeduli Clinic to get medical treatment. Medical treatment at SoloPeduli Clinic is divided into two parts namely outpatient and inpatient health services. Patients generally do not know how long it will take for service from arrival to discharge at the clinic. The queue flow is called a system, while the patients queuing at each service are called states, the number of states cannot be limited as health services must be available 24 hours a day. This issue includes a Discrete Event System which can be modeled with a Petri Net. Petri Net is a mathematical model that can be used to describe certain events in a Discrete Event System. In this research, service flow is modeled using Petri Net which is then represented in the form of a matrix. Coverability Tree was created to define life and deadlock. Scheduling is modeled using Max-Plus Algebra. Service times will be randomly selected to facilitate scheduling and length of service time.

1. Introduction

An example of a queuing system is the queuing system at the Clinic. It is known that health is one of the main needs of society. SoloPeduli Clinic is a health clinic that tries to serve patients as well as possible. Medication at the SoloPeduli Clinic is divided into two parts, namely outpatient and inpatient services. The service system at SoloPeduli Clinic has stages starting from entering the queue until the patient completes the examination. Patients generally do not know the queue flow and how long the service process takes, starting from patient registration to receiving medication.

In this research, the service flow was modeled using the Petri Net which was determined from the service process at the SoloPeduli Clinic, then a service scheduling model was built at the SoloPeduli Clinic using Max Algebra Plus. Furthermore, the model is used to simulate the service scheduling time at SoloPeduli Clinic.

2. Max-Plus Algebra

Max-plus Algebra [1] is a set of \( \mathbb{R}_{\max} \) that is equipped with maximum operations (\( \oplus \)) and addition (\( \otimes \)) and denoted as

\[
\mathbb{R}_{\max} = (\mathbb{R}_{\max}, \oplus, \otimes, \varepsilon, \mathcal{E})
\]

where \( \varepsilon = -\infty \) as the identity element at the maximum operation (\( \oplus \)) and \( \mathcal{E} = 0 \) as the identity element at the addition operation (\( \otimes \)) and \( \mathbb{R}_{\max} = \mathbb{R} \cup \varepsilon \).
Operations on Algebra max plus are the maximum ($\oplus$) and addition ($\otimes$) defined
\[ x \oplus y = \max(x, y) \text{ and } x \otimes y = x + y, \quad \text{for } x, y \in \mathbb{R}_{\max}^+. \]
Furthermore, the max plus Algebraic properties [3], $\forall x, y, z \in \mathbb{R}_{\max}^+$, apply
1. associative, $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ and $(x \otimes y) \otimes z = x \otimes (y \otimes z)$,
2. commutative, $x \oplus y = y \oplus x$ and $x \otimes y = y \otimes x$,
3. distributive, $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$
4. have a zero element, $\forall x \in \mathbb{R}_{\max}^+$, there is $\varepsilon$ such that $x \oplus \varepsilon = \varepsilon \oplus x = x$ ,
5. have a unit element, $\forall x \in \mathbb{R}_{\max}^+$, there is $e$ such that $x \otimes e = e \otimes x = x$ ,
6. have an inverse of the addition, $\otimes$, $\forall x \in \mathbb{R}_{\max}^+$ with if $x \neq \varepsilon$ then there is a single $y$ with $x \otimes y = e$ or there is $-x \in \mathbb{R}_{\max}^+$ such that $x \otimes (-x) = e$,
7. have an absorbent element, $x \otimes \varepsilon = \varepsilon \otimes x = e$.

**Definition 2.1.** For $x \in \mathbb{R}_{\max}^+$ and for all $n \in \mathbb{N}$ with $n \neq 0$,
\[ x^{\otimes n} = x \otimes x \otimes \cdots \otimes x \]
\[ x^{\cdot n} = x + x + \cdots + x \]
\[ x^{\cdot n} = nx. \]

Next, For $x \in \mathbb{R}_{\max}^+$, $e^{\cdot 0} = e = 0$.

**3. Petri net**

Petri net is one of the tools to model discrete event systems. For an event to occur, several conditions must be met first. Information about these events and states is expressed in terms of transitions and places, respectively. Place as input states the conditions that must be met for the transition to occur. After the transition occurs the situation will change.

**Definition 3.1.** Petri net [2] 4-tuple $(P, T, A, w)$ with
- $P$: finite set of places, $P = \{p_1, p_2, \ldots, p_n\}$
- $T$: finite set of transitions, $T = \{t_1, t_2, \ldots, t_n\}$
- $A$: arc set, $A \subseteq (P \times T) \cup (T \times P)$
- $w$: weights function, $w$: $A \rightarrow \mathbb{N}$ finite

Petri net can be described as graph, the point of graph is place and transition. On the Petri net, graphs are allowed to use multiple arcs to connect two equivalent points by giving weight to each arc stating the number of arcs. This structure is known as a multidigraph. The way to run Petri net is to run at least one network token when the transition is activated and this process changes the petrinet. All tokens at the input place are reduced / taken as much as the weight of the arc connecting them.

**3.1. Representation of Petri Net by Matrices**

The meaning of the matrix used in Petri net [5].

The backward and forward incidence matrices are $n \times m$ matrices with the $i$-th row element, the $j$-th column is $A_b(i,j) \equiv w(p_i, t_j)$, $A_f(i,j) \equiv w(t_j, p_i)$

From the backward and forward matrices, we get
\[ A = A_f - A_b \tag{1} \]

One of the uses of the backward incidence matrix is to determine the enabling transition. If $(p_i)$ is an input place of the transition $(t_j)$ $(p_i \in I (t_j))$, the arc from place $(p_i)$ to transition $(t_j)$ is zero because there is no arc connecting it, written $(p_i, t_j) = 0$. So that it can be written in the following vector notation:
\[ x \begin{bmatrix} 1 & \ldots & p_n \end{bmatrix}^{\top} \geq w \begin{bmatrix} 1 & \ldots & p_n \end{bmatrix}^{\top} t \tag{2} \]

From equation (2), the enabling transition can be made by finding a column of the backward incidence matrix that is less than or equal to the state vector, so that it can be written concisely into
\[ x \geq A_b e_j \]

and
\[ X(k + 1) = X(k) + Au, \]  
where \( u \) is a column vector that has as many elements as \( m \), which is obtained from the identity column.

4. Maxplus Algebra model from Petri net with time

Obtained definition \[5\]

\[
\begin{align*}
    a(k) &= v_{a,k} + a(k-1), \quad k = 1, 2, \ldots \text{(k-th customer arrival)} \\
    s(k) &= \max \{a(k), d(k-1)\}, \quad k = 1, 2, \ldots \text{(k-th service time)} \\
    d(k) &= \max \{v_{d,k} + v_{a,k} + a(k-1), v_{d,k} + d(k-1)\} \quad \text{(k-th service time is complete)} \\
    v_{a(k)}: & \text{ length of customer arrival at the k-th time} \\
    v_{d(k)}: & \text{ the length of time the customer is served until the k-th time} \text{ (if a server is down the value is } \varepsilon \text{)} \\
    v_{s(k)}: & \text{ length of time the customer is served again until the k-th time} \text{ (if there is no server down value } \varepsilon \text{)}
\end{align*}
\]  

By using maxplus algebraic notation we get an equation:

\[
\begin{bmatrix} a(k) \\ d(k) \end{bmatrix} = \begin{bmatrix} v_{a,k} \\ v_{d,k} \otimes v_{a,k} \\ v_{d,k} \end{bmatrix} \otimes \begin{bmatrix} c \\ a(k-1) \\ d(k-1) \end{bmatrix}
\]

where \( c \) :

\[ (v_{a,k} \otimes a(k-1)) \oplus (c \otimes d(k-1)) = v_{a,k} \otimes a(k-1) \]

and the initial state \( a(0) = d(0) = 0 \).

5. Service flow

![Service Flow SoloPeduli Clinic](image_url)

Figure 1. Service Flow SoloPeduli Clinic
6. Research method
The method used in this research is the method of literature and primary data collection on SoloPeduli Clinic. The steps in this research are as follows:

- Research Preparation
- Data retrieval
- Data analysis
- Creating a Flow from the data taken
- Creating a Petri net from the flow that have been made
- Determining the Matrix Representation
- Knowing the liveness and deadlocks of each flow and build a coverability tree first
- Building a model using max-plus Algebra from the Petri net that has been built
- Determining the patient service processing time from the simulation using data obtained.
- Conclusion

Figure 2. Research method

7. Main result
Petri net flow is made from the flow of patient services at the SoloPeduli Clinic in Figure 1 as follows:

Figure 3. Petri Net Flow
After getting the Petri net model, then the representation matrix will be searched using equation (1), first looking for a backward and forward matrix. Based on the Petri Net model in Figure 3 there are 11 places and 15 transitions, so that many rows \((n) = 11\) and the number of columns \((m) = 15\). Thus, a forward and backwards incidence matrix with order \(11 \times 15\) will be formed:

\[
A_f = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

so we get the matriks incidence:

\[
A = \begin{bmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\end{bmatrix}
\]

The initial state of Petri Nat from Figure 3 is \(X(0) = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^{T}\). The \(T_0\) transition is enabled because it satisfies \((x_0 \geq A_b(:,0))\). To determine the next state, equation (3) is used where \(u\) states a column vector that has as many elements as \(x\), which are obtained from the identity matrix column. So we get \(u = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^{T}\). Then by combining the matrix \(X(k)\), matrix \(u\) and matrix \(A\) into equation (3) and with \(k = 0\),

\[
X(1) = \begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

This means that the new transition states that are enabled are \(T_0\) and \(T_1\). To determine the next new state, then \(X(1)\) becomes the initial state for \(X(2)\), and so on. The following firing procedures will be carried out \(T_1, T_2, T_3, T_4, T_7, T_9, T_{13}\).

If it is done repeatedly, it can be combined and a coverability tree is created.

5
Furthermore, $V_{a,k}$ is the length of the customer's arrival at the k-th time so it can be written $V_{T_0,k}$ and $V_{d,k}$ is the length of time the customer is served until the k-th time so it can be written, 

$V_{d,k} = V_{T_1,k} + V_{T_2,k} + V_{T_3,k} + V_{T_4,k} + V_{T_5,k} + V_{T_6,k} + V_{T_7,k} + V_{T_8,k}$

Furthermore, queuing models modeled in max plus algebra from equation (6):

$T_0(k) = \text{maks}\{(V_{T_0}(k) + T_0(k-1)), (C + T_{13}(k-1))\}$

$T_{13}(k) = \text{maks}\{(V_{d,k} \otimes V_{a,k} + T_0(k-1)), (V_{d,k}) + T_{13}(k-1)\}$

It is assumed the patient needs time to queue before registering ($V_{T_0,k}$) is 4 minutes and it is assumed that each other service requires a waiting time of about 8 minutes. So that the time needed for patients to leave the service system ($V_{T_{13,k}}$) is estimated at minutes, then obtained time to first patient:

$T_0(1) = \text{maks}\{(4 + T_0(0)), (C + T_{13}(0))\} = 4$

$T_{13}(1) = \text{maks}\{(48 \otimes 4 + T_0(0)), (48 + T_{13}(0))\} = 52$

Assumed for the second patient:

$T_0(2) = \text{maks}\{(4 + T_0(1)), (C + T_{13}(1))\} = 8$

$T_{13}(2) = \text{maks}\{(48 \otimes 4 + T_0(1)), (48 + T_{13}(1))\} = 100$

8. Conclusion

The conclusion that can be drawn from this research is if the arrival time of 0th patient's is 07.00, the arrival of 1st patient's is 07.04 then wait in the registration room for 8 minutes and complete the registration at 07.20. Then the patient must wait for the doctor for 8 minutes and be examined for 8 minutes, so that it is declared outpatient at 07.36. The patient gives a prescription to the pharmacist and queues for 8 minutes, then the patient can take the medicine and complete the administration fee for 8 minutes and the service ends at 07.52.

Acknowledgments

We would like to acknowledge Universitas Sebelas Maret which provide authors chance to present this paper, Siswanto as the supervisor, dan thanks a lot to all committee of KNPMP V 2020.

References

[1] Bacelli F, G. Cohen, G. J. Olsder, and J. P. Quadrat, Synchronization and Linearity, An Algebra for Discrete Event Systems, John Wiley and Sons, New York, 1992.

[2] Cassandras C.G. and Stephanie L., Introduction to Discrete Event Systems, Springer Science Business Media, 2008.

[3] Farlow K. G., Max-plus algebra, Doctoral dissertation, Virginia Tech, 2009.
[4] Mustofani, Dian, Tugas 2 Aljabar Max-Plus : Model Antrian Nasabah Bank dengan Menggunakan Aljabar Petri net, Surabaya : ITS, 2012.

[5] Subiono, Aljabar Min Maks Plus dan Penerapannya ver 3.0.0, FMIPA-Institut Teknologi Sepuluh Nopember, Surabaya, 2015. [Murni dan Aplikasi, Vol. 1, No. 4, hal. 192-206, 2011.

[6] Wattimera dan Freya, Aplikasi Petri Net pada sistem Pembayaran Tagihan Listrik PT. PLN (Persero) Rayon Ambon Timur, Universitas Pattimura, Ambon, 2011.

[7] Widayanti, Dwina Nur and Subiono, Penjadwalan Pelayanan di PLN dengan Menggunakan Petri Net dan Aljabar Maks Plus, Prosiding Seminar Nasional FMIPA Universitas Negeri Surabaya ISBN : 978-602-17146-0-7, (B-33)-(B-39).