

On Minimal Crisp Boolean Petri Net generating Boolean Petri Net

Riddhi Jangid^{1†}, Gajendra Pratap Singh^{1*}, Kamalendra Kumar^{2†}

¹School of Computational and Integrative Sciences, Jawaharlal Nehru
University, New Delhi, 110067, Delhi, India.

¹Department of Basic Science, Shri Ram Murti Smarak, College of
Engineering and Technology , Bareilly, 243202, UP, India.

*Corresponding author(s). E-mail(s): gajendra@jnu.ac.in;
Contributing authors: riddhi40_sit@jnu.ac.in;
kamlendra.14kumar@gmail.com;

[†]These authors contributed equally to this work.

Abstract

Petri net (PN) is a graphical modeling tool used to learn the behaviour of a discrete-event type dynamical system using one of its analysis technique, Reachability. Boolean Petri Nets(BPNs) is one such variant in PNs that generate every possible binary n-vector in their Reachability analysis. It is used to study the safe systems that can be applied to optimize resource allocations, database systems, etc. The problem of generating binary vectors is limited only to a certain defined Petri Net graph. We extend the concept of BPNs by studying Crisp BPNs and its reachability analysis. We initialize by constructing BPNs from the Minimal Crisp BPNs and adding on some more Nets in BPNs that generate all possible binary vectors as a subset of the marking vectors in their reachability.

Keywords: Marking vectors, Firing rule, Boolean and Crisp Boolean Petri nets, Reachability Tree

2000 Mathematics Subject Classification: 05C90 , 68R01 , 93C29 , 97N70

SUBMISSION DATE: 25.10.2024

1 Introduction and Preliminaries

Mathematical modeling using Petri Nets offers good visualization and understanding of structurally and behaviorally discrete dynamic systems. From many of the graph theoretical[9, 13] concepts, Petri Net(PN) is one graphical modeling tool that was introduced during 1960s and has been used extensively since then in many areas of science[17–19, 22]. Several new types of Petri Nets(PNs) have been introduced depending on the systems' modeling requirements[1]. Meanwhile, Boolean Petri Nets [4] (Abbreviated as BPN) came into the picture that corresponds to the class of PNs that generate all binary marking vectors at any stage of its reachability[5]. Their application is in studying the safe systems that include all possible binary n-vectors [8, 24]. This study introduced a novel idea of Crisp Boolean Petri Nets [11], which has a count of one of each of these binary n-vectors during their reachability analysis. The authors here show their interest in studying the Crisp BPN, and their analysis and conclude with new results in the theory. For this purpose, initializing by introducing Crisp BPNs as defined by [15] and then constructing Star PN using the Crisp BPN is shown in section 2. The motivation behind studying the Minimal Crisp BPNs is that they generate all the binary n-vectors in their first level of reachability depicting the topology of a star graph. Converting minimal Crisp BPNs into BPNs will generate all the binary vectors of higher dimensions more quickly than in the minimum and minimal Crisp Boolean Petri Nets[7, 15, 16]. We have analyzed the constructed Net and concluded with the results in the next sections.

Though researchers define Petri Nets differently[2, 20], their idea remains intact. In the manuscript, we follow Murata's [6] definition of Petri Net i.e. $N = (P, T, F, W, \mu_0)$. Here P (Places) and T (Transitions) represent two disjoint finite set of nodes of a graph, F represents the set of directed arcs between the nodes and W represents the 'weight function' mapped from the set of directed arcs to the non-negative integers. Graphically we denote the set P using circles and the set T using rectangles. Petri Net graph is a representation of the PN that is a 'directed bipartite multigraph' in which the node set of the graph is $P \cup T$, the arc set is F , and μ_0 is the k -vector associated with P that represents the initial marking of the graph where $k = |P|$. The non-negative integer values in the k -vector are count of tokens in the places of the Net. 'Tokens' are associated the set P and denoted using dots inside the circles graphically. With the graph, we associate events in any discrete system that takes place based on the conditions of that system. This behavior is modeled using the execution rule of the PN. This rule is divided into two parts:

Enabling condition for a transition:

1. Enabled transition means that it contains at least the count of tokens in each of its input places as the weight of the incident arcs from the place to the transition.
2. Firing of transition: Any transition fires when $w(p, t)$ number of tokens from each of the respective input places of that transition is removed and $w(t, p)$ number of tokens are added to each of the output places of the transition. This changes the current state of the PN. ' w ' here denotes the weight of the respective arc from p to t or vice-versa.

Execution Rule in a Petri Net: PN executes by firing of transitions; that transition must be enabled for firing of a transition. Also not every enabled transition fires, it depends upon the system requirements. This execution gives us different states of the

Petri Net. Fig. 1 explains a Petri Net and shows the transition firing that results in the different states of the Net.

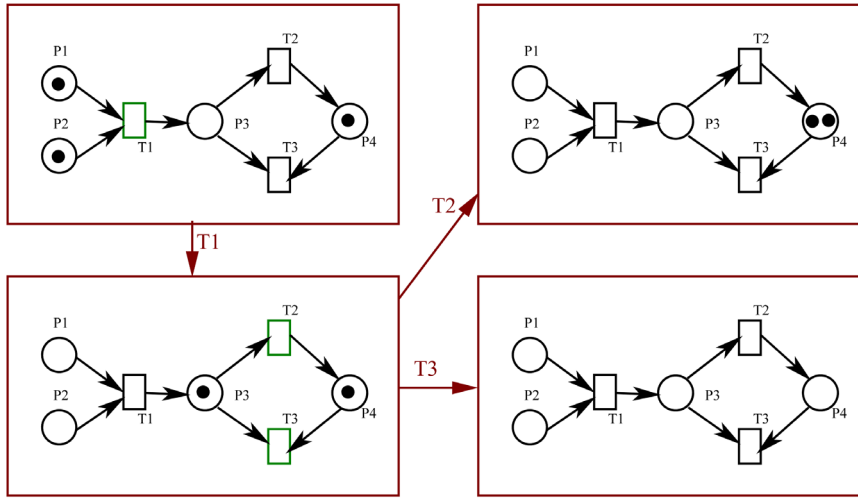


Fig. 1 Illustrative example of firing in a PN with $\mu_0 = (1, 1, 0, 1)$

The structural and behavioral analysis of a PN is using Matrix Analysis and Reachability Analysis techniques. We adopt the latter in the work since it is more expressible, easy to understand, and comparatively quick to analyze any net.

Any system needs to function correctly after its design and that is where we encounter the problem of reachability. We are concerned about whether the designed system is able to reach some particular state/marking or not. If yes, what can be the firing sequence that gives us that marking? This is stated in the reachability problem [21]. Hence, any marking μ_i is reachable from the initial marking μ_0 whenever a transition firing sequence (say σ_i) exists. Formally, $\mu_i \in R(N, \mu_0)$ where $R(N, \mu_0)$ denotes the set of all reachable markings of the PN N and is known as the reachability set. Sometimes, it is also possible that we reach μ_i with more than one firing sequence which gives us the different functional behavior of the system to reach a particular state.

In Fig. 2 μ_1, μ_2, μ_3 are the reachable markings from the initial marking μ_0 . Evaluating from Fig. 1, $\mu_0 = (1, 1, 1, 0)$ and μ_1, μ_2, μ_3 equals $(0, 0, 1, 1), (0, 0, 2, 0), (0, 0, 0, 0)$ respectively in Fig. 2. The green color of μ_0 in Fig. 2 depicts the initial state in the PN while the green color of transitions in Fig. 1 shows that the transitions are enabled in the respective state of the PN.

Conflict and parallelism in PNs: Consider a situation in a PN in which more than one transition having at least one common place is enabled such that the firing of one transition will disable another transition in a PN. In that case, which transition will fire first? This situation created in a PN is the situation of conflict. There may be events that require the firing of the first transition while some may require the firing

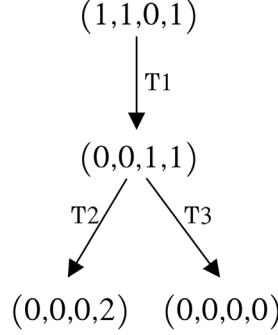


Fig. 2 Reachability analysis of PN shown in Fig.1

of other transitions. Hence, while modeling it is an important aspect that needs to be taken care of. On the other hand, some events can work independently in a given system, i.e. there is no common place among the enabled transitions, and the firing of one transition will not affect the other transition. These enabled transitions can work independently in a PN. Such a situation in a Petri Net is termed as Parallelism in the Petri Net. These terms are useful while structural modeling of a system in a Petri Net.

2 Construction of Source Petri Net

For constructing a Source Petri net, we use the concept of Boolean Petri nets(BPNs). Boolean Petri Nets(BPNs) belong to a class of Petri Nets that follows all the definitions of a PN but differ only in the reachable markings of a PN. Simplifying the conventional definition of BPNs, the markings in the reachability analysis of these PNs are all possible marking vectors of a Boolean hypercube [6]. If PN is Boolean then the initial marking vector of the Net is $(1, 1, \dots, 1)$, and the vectors in its reachability tree are all possible combinations of 0 and 1 and out of which 2^k are unique, where k is the number of places on the Net. The reason why BPNs have been chosen in the study is because they are used to study safe systems in many applications like control systems. More refinement in the concept of Boolean Petri nets is studied using Crisp Boolean Petri Nets. Crisp Boolean Petri Nets[15] correspond to those Boolean Nets that generate 2^k Boolean marking vectors exactly once in their reachability i.e. $|R(N, \mu_0)| = 2^k$.

Fig. 3 shows a generalized Crisp Boolean Petri Net where $|P| = k$ and $|T| = 2^k - 1$. Interestingly, the underlying graph of the reachability tree of any Crisp BPN is isomorphic to a Star graph $K_{1,2^k-1}$. As an example, the reachability tree of a Crisp Boolean Petri Net is shown in Fig. 4 One can refer to [11, 12] for a complete study concerning the topic. In the paper, the authors show interest in the Star graph property of the reachability graph of the Crisp BPN. We construct a Petri Net using the directed graph of the reachability tree of minimal Crisp Boolean Petri Net.

The reachability tree generated while analyzing the Crisp Boolean Petri Net is isomorphic to the graph $K_{1,2^k-1}$ with all of its arcs directed downward.

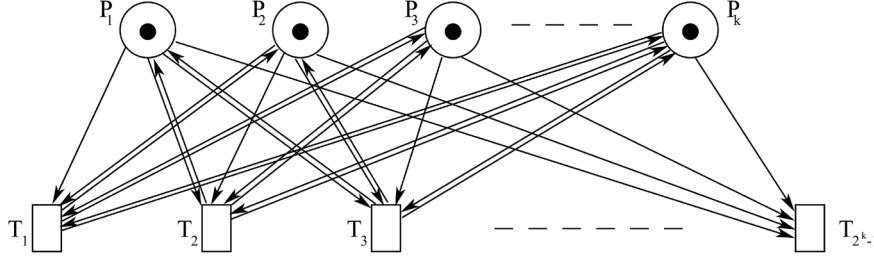


Fig. 3 Crisp Boolean Petri Net with k places

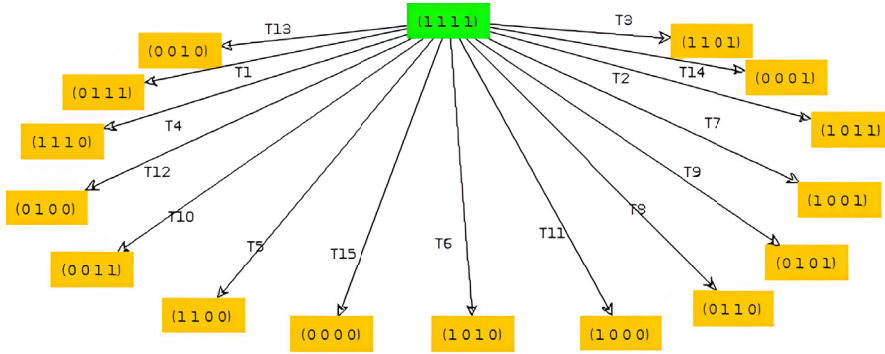


Fig. 4 Reachability tree of Crisp Boolean Petri Net with $k = 4$ places

2.1 General construction steps for Source Petri Net

It is to keep in mind that for any number k of the places in the Crisp Boolean Petri Net, the markings generated in the reachability tree are 2^k (including the initial marking vector). We subdivide $K_{1,2^k-1}$ Star graph to construct the Boolean Petri Net of order $2^k - 1$. The general construction steps for creating a Source Petri Net are given as-

Step 1. Replace the marking nodes of the tree with the transitions, labeling each marking such that every unique Boolean marking vector μ_i is labeled as T_i (including root node) where $0 \leq i \leq 2^k - 1$ and $i \in \mathbb{Z}$.

Step 2. Subdivide every arc $(T_0, T_i) \forall i$ by a place node and label every corresponding place as P_i where $1 \leq i \leq 2^k - 1$ and $i \in \mathbb{Z}$. (It is important to note that labeling should be assumed similar for both the places and corresponding transitions without any loss of generality. For eg. $T_0 \rightarrow P_2 \rightarrow T_2$.)

We call this constructed Petri Net as a Source Petri Net of order $(2^k - 1)$ since it contains a source transition associated with each of its places.

As an example, the Petri Net constructed using Star reachability tree (isomorphic to $K_{1,7}$) of Crisp Boolean Petri Net with $k = 3$ is shown in Fig. 5 and we call it Source Petri Net of order 7.

From the construction, we infer that the number of transitions in the defined Source Petri Net will always be 2^k and the number of places will be $2^k - 1$ using the fact that if a Star graph contains n number of vertices then it has $n - 1$ number of edges.

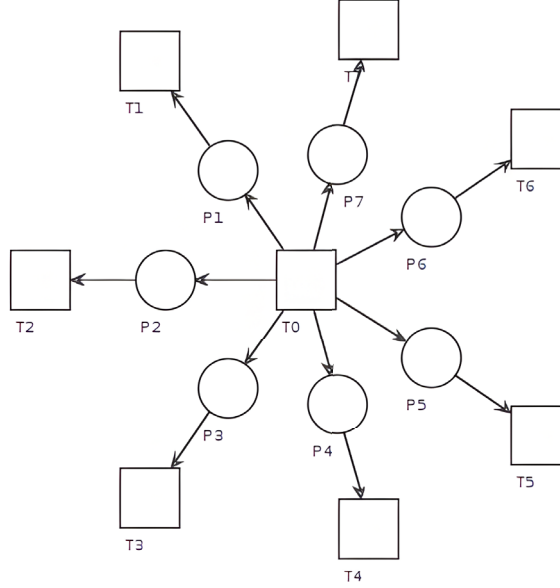


Fig. 5 Source Petri Net of order $2^k - 1$ constructed for $k = 3$

2.1.1 Artificial place

The place introduced in the model is just to limit the firing ability of a source transition and does not contribute to the marking of the Petri Net. We define it as an Artificial place. For instance, P_0 is the Artificial place in the Fig. 6 depicted using a dotted outline that has no role in marking of the net.

3 Analysis and Results

In this section, we characterize the constructed Source Petri Net and dig out some useful information out of it. Characterization plays an important role in understanding a system as discussed in [23]. Since we have already discussed that the transition T_0 is a source transition for every place in the Petri Net while all other transitions are sink transitions. This results in an infinite reachability tree of the Petri Net generating every possible marking vector. The reachability tree is a complete ocean that depicts all the possible states of the system in every possible firing of transitions. Since, source transition is the reason behind creating the infinite reachability tree, we need to restrict the infinite firing of this transition and to deal with the concern, different cases can be considered.

3.1 Case I

Reversing the direction of arcs from T_0 to P_i in the Source Petri Net and assigning initial marking 1 to every place P_i for $0 \leq i \leq 2^k - 1$ to make a finite reachability tree for analysis. Interestingly, this construction gives us Petri Net isomorphic to 1-safe

Star Petri Net S_n in Boolean Petri Nets that has already been discussed in [4] (Refer Figure 1. in [4]) Below is the proposition that has been discussed as a part of the case.

Proposition 1. *Source Petri Net for $2^k - 1$ places, after reversing all the arcs from T_0 to P_i , is isomorphic to the 1-safe Star Petri Net S_n ([4]) corresponding to its order.*

Proof. Let us consider the Source Petri Net graph constructed using Crisp Boolean Petri Net of k places. As suggested in Case I, we reverse the direction of arcs from T_0 to P_i in the Source Petri Net and assign initial marking 1 to every place P_i for $0 \leq i \leq 2^k - 1$. The Petri Net graph contains $2^k - 1$ places and 2^k transitions. The corresponding order of places and transitions of 1-safe Star Petri Net S_n contains $n - 1$ places and n transitions. This implies $n \sim 2^k$. We need to prove that both these graphs are isomorphic.

Number of places and transitions in the graphs are same for $n = 2^k$. Sufficient conditions to prove isomorphism will be the same adjacency matrix for the corresponding undirected graphs and oriented in the same direction. Another way to prove isomorphism is by showing one-on-one correspondence between the graphs. We use the latter one.

Map the place nodes P_i (representation of place in Source Petri Net graph) with p_i (representation of place in 1-safe Star Petri Net S_n) for every $n = 2^k$ and $i \in N$. Similarly, map the transition nodes T_i (representation of transition in Petri Net graph constructed using Crisp Boolean Petri Net) with t_i (representation of transition in 1-safe Star Petri Net S_n) for every $n = 2^k$ and $i \in N$ and $T_0 \sim t_{n+1}$. This proves isomorphism between the graphs. $\square$

3.2 Case II

Removing the central transition T_0 and its associated arcs and assigning initial marking 1 to every place P_i for $0 \leq i \leq 2^k - 1$ for finiteness in the reachability analysis. This can be studied as a subcase of the work discussed in [3] that generates hypercube and mesh networks as an underlying graph of their reachability. Hypercubes and other network topologies are studied in different aspects also, that is a different area of research. [10]

3.3 Case III

Adding an artificial place P_0 to the source transition T_0 such that the transition is not a source transition anymore (Refer Fig. 6). In this case, we do not need to add tokens to every place instead we can add a token to P_0 and our purpose is solved. The rule we will follow is, as soon as T_0 fires remove the artificial place P_0 . This can also be thought of as the firing of transition T_0 only once when it is a source transition. For $k = 3$, the resulting Petri Net after firing T_0 and removing P_0 is the same as in Fig. 5 but with initial marking (1, 1, 1, 1, 1, 1, 1).

We consider here the third case and denote this class of Petri Nets as Flying Swing Petri Net ($FSPN$) that is dependent on the value of $k \in N$ in Crisp Boolean Petri Net.

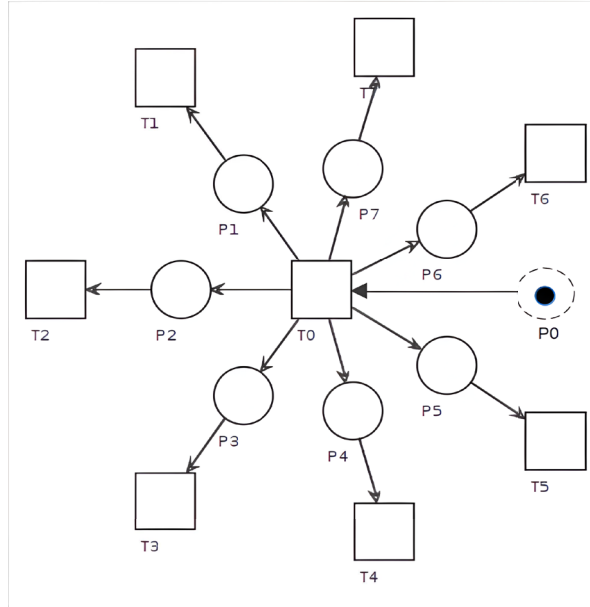


Fig. 6 Source Petri Net of order $2^k - 1$ constructed for $k = 3$

3.3.1 Flying Swing Petri Net

Flying Swing Petri Net ($FSPN$) can be defined as 5-tuple $FSPN = (P, T, F, W, \mu_0)$ where,

$$\begin{aligned} P &= \{P_1, P_2, \dots, P_{2^k-1}\}, \\ T &= \{T_0, T_1, \dots, T_{2^k-1}\}, \\ F &\subseteq (P \times T) \cup (T \times P), \\ W &: F \rightarrow \{1\}, \\ \mu_0 &= (0, 0, 0, \dots, 0). \end{aligned}$$

such that T_0 remains the central transition acting as an input transition for all the places and rest remain sink transitions with exactly one input place.

NOTE: It is important to note that the place P_0 is an Artificial place that has no role in the markings of the Net i.e. $P_0 \notin P$. It is introduced to limit the firing of transition T_0 .

Theorem 2. Every binary $(2^k - 1)$ -vector is generated in the reachability analysis of Flying Swing Petri Net $FSPN = (P, T, F, W, \mu_0)$ where $|P| = 2^k - 1$, $|T| = 2^k$ with initial marking $\mu_0 = 0 \forall P_i \in P$ and $i, k \in \mathbb{N}$.

Proof. We prove this using Mathematical Induction.

We know that the total number of unique binary k -vectors is 2^k . For $k = 1$, the total binary vectors are $2^1 = 2$. These are specifically 0 and 1. The $FSPN$ constructed for $k = 1$ contains $2^1 = 2$ transitions (T_0, T_1) and $2^1 - 1 = 1$ place (P_1) such that $T_0 \rightarrow P_1 \rightarrow T_1$. One time firing of T_0 using an artificial place P_0 results in the marking (1)

and further firing of T_1 results in the marking (0) . Hence, $R(FSPN, \mu_0) = \{(0), (1)\}$ and $|R(FSPN, \mu_0)| = 2$ where $\mu_0 = (0)$. Hence, the results hold true for $k = 1$.

For $k = 2$, the total binary k -vectors are $2^2 = 4$. These are specifically $(0, 0)$, $(1, 0)$, $(0, 1)$ and $(1, 1)$. The $FSPN$ constructed for $k = 2$ contains $2^2 = 4$ transitions (T_0, T_1, T_2, T_3) and $2^2 - 1 = 3$ places (P_1, P_2, P_3) . One time firing of T_0 using an artificial place P_0 results in the marking $(1, 1, 1)$ and further firing of T_1 , T_2 and T_3 results in the markings $(0, 1, 1)$, $(1, 0, 1)$ and $(1, 1, 0)$ respectively. This continues until all the transitions are dead in the $FSPN$. The reachability graph is shown in Fig. 7. It clearly states that $R(FSPN, \mu_0) = (0, 0, 0), (0, 0, 1), (0, 1, 0), (1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)$ and $|R(FSPN, \mu_0)| = 2^3 = 8$ where $\mu_0 = (0, 0, 0)$. Hence, the results hold for $k = 2$.

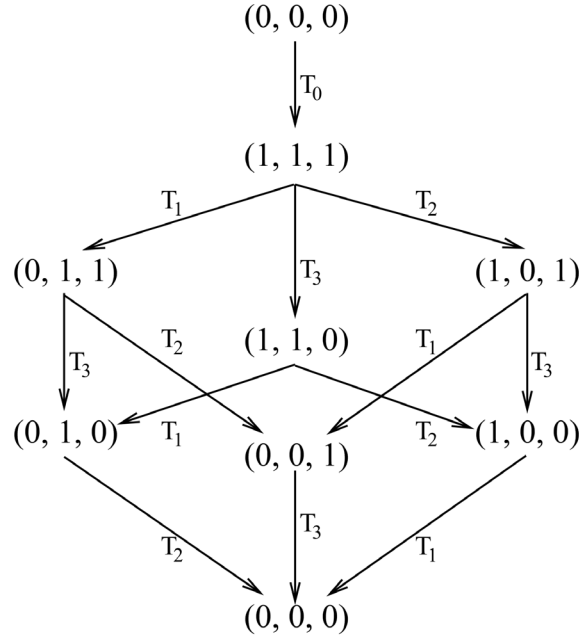


Fig. 7 Reachability graph constructed for $FSPN$ with $k = 2$

Let us suppose that the result holds true for $k = r$. This implies $\mu_i = (\mu_i(P_1), \mu_i(P_2), \dots, \mu_i(P_{2^r-1})) \in \{0, 1\}^{2^r-1}$, i.e. total number of unique binary vectors are 2^{2^r-1} in $FSPN$ with $2^r - 1$ places, 2^r transitions and $|R(FSPN, \mu_0)| = 2^{2^r-1}$ where $\mu_0 = (0, 0, 0, \dots, 0)$. Now, for $k = r + 1$, the number of places in $FSPN$ will be $2^{r+1} - 1$ and number of transitions will be 2^{r+1} . We must calculate the number of binary vectors in the reachability analysis of such $FSPN$. For the marking vectors, initially, we require the difference in the number of places $(2^{r+1} - 1) - (2^r - 1) = 2^r$. Hence, we have the marking vector $\mu_i = (\mu_i(P_1), \mu_i(P_2), \dots, \mu_i(P_{2^r-1}), \dots, \mu_i(P_{2^{r+1}-1}))$ as an intermediate state in $FSPN$ with $2^{r+1} - 1$ places and 2^{r+1} transitions.

Consider the marking vector $\mu_i = (\mu_i(P_1), \mu_i(P_2), \dots, \mu_i(P_{2^r-1}), \mu_i(P_{2^r}))$. We augment the markings $\mu_i = (\mu_i(P_1), \mu_i(P_2), \dots, \mu_i(P_{2^r-1})) \in \{0, 1\}^{2^r-1}$ and $\mu_i(P_{2^r})$ since for $\mu_i(P_{2^r})$ we have two possible markings, 0 or 1. Hence, the total possible binary vectors will be $2^{2^r-1} \cdot 2 = 2^{2^r}$. This implies $\mu_i = (\mu_i(P_1), \mu_i(P_2), \dots, \mu_i(P_{2^r})) \in \{0, 1\}^{2^r}$.

Proceeding in the same way for $\mu_i = (\mu_i(P_1), \mu_i(P_2), \dots, \mu_i(P_{2^r})) \in \{0, 1\}^{2^r}$ and $\mu_i(P_{2^r+1})$, total binary vectors will be 2^{2^r+1} . Hence, continuing in the same way for 2^r more places in the Net, we have 2^r transitions associated with them that will fire creating every possible marking that is binary i.e. total number of binary vectors for the marking $\mu_i = (\mu_i(P_1), \mu_i(P_2), \dots, \mu_i(P_{2^r-1}), \dots, \mu_i(P_{2^{r+1}-1}))$ will be $2^{2^r-1} \cdot 2 \cdot 2 \dots 2 = 2^{2^r-1+2^r} = 2^{2^{r+1}-1}$. This implies $|R(FSPN, \mu_0)| = 2^{2^{r+1}-1}$ for $FSPN$ with $2^{r+1} - 1$ places and 2^{r+1} transitions as the required result.

Hence by mathematical induction, we can say that every binary $(2^k - 1)$ -vector is generated in the reachability analysis of Flying Swing Petri Net $FSPN = (P, T, F, W, \mu_0)$ where $|P| = 2^k - 1$, $|T| = 2^k$ with initial marking $\mu_0 = 0 \forall P_i \in P$ and $i \in N$. $\square$

Why choose this model? The model constructed is a safe persistent net that can also be transformed into a marked graph [14]. We consider here an example of the theorem discussed in this section to validate the results. Let us assume $k = 3$. Hence, $FSPN = (P, T, F, W, \mu_0)$ where $|P| = 2^3 - 1 = 7$, $|T| = 2^3 = 8$ with initial marking $\mu_0 = (0, 0, 0, 0, 0, 0, 0)$. Construction of Source Petri Net for $k = 3$ is the same as shown in Fig. 5. We are initiating the firing of transition T_0 in the Petri Net using an artificial place results in the marking $(1, 1, 1, 1, 1, 1, 1)$. We need to validate the result by generating all the possible binary 7-vectors in its reachability. We have used Python3 for the reachability and validated our result as shown in Fig. 8. It depicts to us the number of unique reachable nodes we have generated is equal to $|R(FSPN, \mu_0)| = 2^7 = 128$.

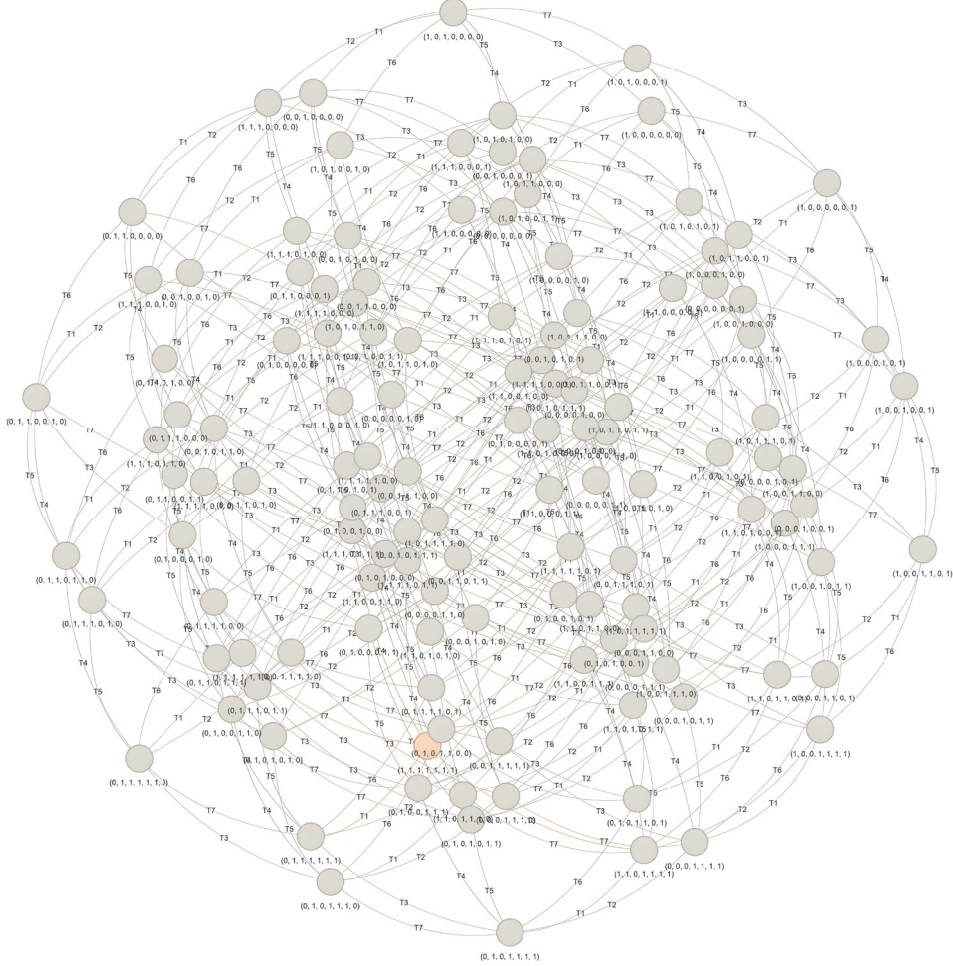


Fig. 8 Reachability graph constructed for $FSPN$ with $k = 3$

4 Conclusion and Discussion

The concept of Boolean Petri Nets has been explored from different perspectives and different methods to analyze. Crisp Boolean Petri Nets are of special interest because of their uniqueness in marking vectors of their reachability. We project here different possible cases to generate all the binary vectors in its analysis highlighting the previous work contributed in this direction. The results have been validated but the state space explosion in reachability is another problem here difficult to deal with. One other limitation is that we have the generated Boolean Petri Nets for some specific order only. Other marking vectors than of order $(2^n - 1)$ are not generated here due to focus on the generation of boolean marking vectors only. This gap can be explored

when focusing on the number of places in the net. Researchers are motivated to look into the problem and work towards this direction.

Acknowledgements. The second author, GPS is thankful to the Department of Science and Technology (DST)-Science and Engineering Research Board (SERB) project (MATRICS Id: MTR/2021/000378), ICMR project ID 2021-15450, M.Sc consumable grant, DBT BIC and the Department of Biotechnology, Ministry of Science & Technology, Govt. of India (Project id BT/PR40251/BITS/137/11/2021) to provide the research facility. The authors are expressing their deep gratitude to anonymous reviewers, and editors for their valuable suggestions and comments.

Statements and Declarations

- Competing interests
All the authors have no competing interests.
- Ethics approval
This research is an original work of the authors and has not been previously published elsewhere.
- Consent for publication
All the authors have given their consent for publication.
- Data availability
Not applicable

References

- [1] Gupta, S., Kumawat, S., & Singh, G. P. (2019). Fuzzy Petri net representation of fuzzy production propositions of a rule based system. In *Advances in Computing and Data Sciences: Third International Conference, ICACDS 2019, Ghaziabad, India, April 12-13, 2019, Revised Selected Papers, Part I 3* (pp. 197-210). Springer Singapore.
- [2] Gupta, S., Singh, G. P., & Kumawat, S. (2019). Petri net recommender system to model metabolic pathway of polyhydroxyalkanoates. *International Journal of Knowledge and Systems Science (IJKSS)*, 10(2), 42-59.
- [3] Jangid, R., & Singh, G. P. (2023). Construction of m-cube mesh Networks using Petri Nets. *International Journal of System Assurance Engineering and Management*, 1-7.
- [4] Kansal, S., Acharya, M., & Singh, G. P. (2012). Boolean Petri Nets. *Petri Nets Manufacturing and Computer Science* (Ed.: Pawel Pawlewski), 381-406.
- [5] Kansal, S., Singh, G. P., & Acharya, M. (2010). On Petri nets generating all the binary n-vectors. *Scientiae Mathematicae Japonicae*, 71(2), 209-216.

- [6] Kansal, S., Singh, G. P., & Acharya, M. (2011). 1-Safe Petri nets generating every binary n -vector exactly once. *Scientiae Mathematicae Japonicae*, 74(1), 29-36.
- [7] Kansal, S., Singh, G. P., & Acharya, M. (2011, July). A disconnected 1-safe petri net whose reachability tree is homomorphic to a complete Boolean lattice. In 2011 International Conference on Process Automation, Control and Computing (pp. 1-5). IEEE.
- [8] Kansal, S., Singh, G. P., & Acharya, M. (2015). On the problem of characterizing Boolean Petri Nets. *International Journal of Computer Applications*, 975, 8887.
- [9] Kumar, K., & Panigrahi, P. (2025). Antipodal Number of Cartesian Product of Complete Graphs with Cycles. *Communications in Combinatorics and Optimization*, 10(1), 219-231.
- [10] Mirafzal, S. M. (2023). On the distance-transitivity of the folded hypercube. *Communications in Combinatorics and Optimization*.
- [11] Murata, T. (1989). Petri Nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4), 541-580.
- [12] Peterson, James L. Petri Net theory and the modeling of systems. NJ: Prentice-Hall, Inc., Englewood Cliffs, 1981.
- [13] Senthilkumar, B., Chellali, M., & Yanamandram, V. B. (2023). Graphs with unique minimum edge-vertex dominating sets. *Communications in Combinatorics and Optimization*.
- [14] Singh, G. P., & Kansal, S. (2016). Basic results on Crisp Boolean Petri Nets. In *Modern Mathematical Methods and High Performance Computing in Science and Technology: M3HPCST*, Ghaziabad, India, December 2015 (pp. 83-88). Springer Singapore.
- [15] Singh, G. P., Kansal, S., & Acharya, M. (2013). Existence and uniqueness of a minimum crisp Boolean Petri Net. *International Journal of Computer Applications*, 73(20).
- [16] Singh, G. P., Kansal, S., & Acharya, M. (2013). Construction of a crisp Boolean Petri net from a 1-safe Petri net. *International Journal of Computer Applications*, 73(17), 1-4.
- [17] Singh, G. P., & Jha, M. (2022). Modeling the Immune Response of B-Cell Receptor Using Petri Net for Tuberculosis. In *Advanced Computational Techniques for Sustainable Computing* (pp. 259-272). Chapman and Hall/CRC.

- [18] Singh, G. P., Jha, M., & Singh, M. (2021). Applications of Petri Net Modeling in Diverse Areas. In *Mathematical Modeling, Computational Intelligence Techniques and Renewable Energy: Proceedings of the First International Conference, MMCITRE 2020* (pp. 437-449). Springer Singapore.
- [19] Singh, G. P., Jha, M., Singh, M., & Naina. (2020). Modeling the mechanism pathways of first line drug in Tuberculosis using Petri nets. *International Journal of System Assurance Engineering and Management*, 11, 313-324.
- [20] Singh, G. P., & Singh, S. K. (2019). Petri net recommender system for generating of perfect binary tree. *International Journal of Knowledge and Systems Science (IJKSS)*, 10(2), 1-12.
- [21] Singh, G. P., Singh, S. K., & Jha, M. (2020). Existence of forbidden digraphs for Crisp Boolean Petri Nets. *International Journal of Mathematical, Engineering and Management Sciences*, 5(1), 83.
- [22] Singh, M., Singh, G. P., & Tyagi, S. (Eds.). (2022). *Microbial Products: Applications and Translational Trends*. CRC Press.
- [23] Singh, S. K., Jangid, R., & Singh, G. P. (2023). On characterizing binary Petri Nets. *International Journal of System Assurance Engineering and Management*, 14(3), 919-929.
- [24] Steggles, L. J., Banks, R., & Wipat, A. (2006). Modelling and analysing geNetic Networks: From Boolean Networks to Petri Nets. In *Computational Methods in Systems Biology: International Conference, CMSB 2006, Trento, Italy, October 18-19, 2006. Proceedings* (pp. 127-141). Springer Berlin Heidelberg.